WWW Proxy Traffic Characterization with Application to Caching

Ghaleb Abdulla, Edward A. Fox, Marc Abrams, Stephen Williams

Abstract

Characterizing World Wide Web proxy traffic helps identify parameters that affect caching, capacity planning and simulation studies. In this paper we identify invariants that hold across a collection of ten traces representing traffic seen by caching-proxy servers. The traces were collected from governmental, industry, university, high school, and an online service provider environment, with request rates that range from a few accesses to millions of accesses per hour. We also show that the examined traffic is self-similar. We explore sources of Web self-similarity and we conclude that a strong source is the periodicity in the users behavior. The tests revealed that their is a strong connection between access rate from hour to hour. We also report the hit rate and weighted hit rate obtained by running a trace driven simulation on the workloads to simulate a proxy with infinite cache. similarly, accesses to unique servers and URLs are a small portion of the total. By considering these characteristics of traffic we can improve the utility of caching for WWW clients.

1 Introduction

Research to reduce network latency, enhance response time and conserve network bandwidth when accessing a World Wide Web document is very active. Simulations using server traffic [1, 2, 3] and proxy traffic [4, 5] have shown that caching will reduce latency and network usage. Understanding Web traffic is essential for performing and generalizing results of such simulations and experiments. It is important to look for workload characteristics that will hold true between different workloads.

The dynamics of Web traffic are not well understood. There are several differences between the Web and other types of network traffic. Those differences emerge from the protocol used and Web users behavior. With respect to the protocol, traffic is generated by clicking on hyperlinks that are part of HTML pages and, as a result, a new HTML page or an image is displayed. HTML pages contain formatted text and nicely organized graphics. Sometimes HTML pages lead to other types of media, such as video or audio. In contrast, traditional network traffic has formatted or unformatted text, and rarely uses graphics, video or audio. With respect to users, the lower level of expertise required to navigate with a Web browser has resulted in a large and diverse user population. Therefore, it is reasonable to assume that Web users behave differently from those who use other network resources. Change in user behavior over a period of time may affect the Web dynamics too.

1.1 Related work

Their were several studies to characterize client workload [6, 7] and server workload [1]. However, their was no study to characterize proxy traffic. This is due to the difficulty of collecting proxy log files from different sources due to the sensitivity of such logs. In [1] the authors used six different log files to characterize accesses to WWW servers. From the used logs the authors identified ten differnt invariants for web server workload. The invariants are important since they represent universal truths for all Internet Web servers. The invarinats in the study were used to identify two strategies for cache design and to determine the bounds on performance improvement due to each strategy. A study to test if those invariants hold true over time is neccessary.

In [6] and [7] the data was collected from a group of clients accessing the Web. The authors [7] in showed that many characteristics of the WWW can be modeled using power-law distributions. In [6] the authors have shown that the Web traffic have characteristics that is consistent with self-similarity. They traced the raesons for Web self-similarity to the basic characteristics of information organizaion and retrieval.

In this paper we characterize the traffic seen by a caching-proxy by identifying a set of invariants that hold true between the examined workloads. We also test for the existence of self-similarity in the traffic and the reasons for it. We examine caching-proxy traffic over time and see if the identified invariants change over time. Finally, we run a trace driven simulation for all workloads and report the hit rate and weighted hit rate for each workload.

2 Objectives

In this paper we test the following hypotheses:

- Hypothesis 1: We can identify invariants across proxy workloads collected from different communities, as was done for server workloads in [1].
- Hypothesis 2: Some of the identified invariants hold true over time.
- Hypothesis 3: The traffic seen by a proxy is self-similar.
- Hypothesis 4: A major source for the traffic self-similarity is the cyclic behavior of users over time.

We test these hypotheses by examining ten different workloads that were collected from different environments; this will allow generalizing results and conclusions from this paper.

Workload	Period	# of accesses	Bytes transferred(MB)
DEC1	Sep/3/1996	1304565	11206.99
DEC2	Sep/19/1996	1293147	10889.42
BU(G)	Nov/29/1994-Feb/27/1995	52901	293.99
BU(U)	Jan/27-Feb/22/1995	414350	1201.18
Korea	Sep/02-Sep/26/1995	1681963	21941.65
VT-Lib	Sep/19-Nov/20/1996	127853	589.21
VT-CS	Jan/1-Nov/18/1996	570385	3491.74
VT-Han	Jul/12-Nov/20/1996	440345	2577.67
AUB	Oct/21-Oct/22/1996	19259	109.52
AOL	Dec/96(few minutes)	883082	6017.88

Table 1: Workloads used in this study.

The rest of this paper is organized as follows. Section 3 describes the workloads used in the study. Section 4 identifies invariants that hold across the workloads. Section 5 discusses the invariants over time, and it presents analysis of correlation between access rate for different time scales. Section 6 discusses the effect of the identified invariants on hit rate and weighted hit rate. Finally, section 7 presents the conclusions and future work.

3 Workloads Studied

To generalize our results we collected data from governmental, industry, university, high school and an online service provider sources. Table 1 summarizes the workloads used in this study, showing dates of collection. For each workload we have collected a log file that contains users' accesses to the WWW.

Collection procedures differ between workloads. Workloads from Virginia Tech, Computer Science (VT-CS), Hancock Hall(VT-Han), and Library (VT-Lib) and Auburn (AUB) high school were collected using a tool called *httpfilt* [8]. Since the tool records all accesses from the clients to servers inside and outside the school, we used a filter to exclude accesses to local servers. This way we only examine accesses to servers outside the school which is what a caching-proxy would see. The Digital workloads DEC1 and DEC2 were collected using a modified version of the 1.0.beta17 squid proxy. The modified proxy was installed on two machines that act as Web proxies for Digital's internal network. The Boston University log files (BU(G) for graduate students and BU(U) for undergraduates) were collected by modifying a version of Mosaic, which was popular at the time of collection, to record certain information for each client. America On Line (AOL) trace also was collected using a proprietary proxy server; however their log file had only the URLs accessed by users and it did not have clients, sizes, or timing information for privacy reasons. Using the available URLs and a locally developed software tool called *WebJama* we replayed all AOL accesses through a modified Harvest server and generated a new log file which includes file size information. The Korea log file was collected using a proxy server installed as the gateway to South Korea. Therefore, the logs contain all trans-Pacific traffic.

4 Proxy Workload Invariants

In this section we identify invariants that hold true across the workloads studied. We follow closely the work done in [1] in establishing the invariants. However, since our workloads are for a different class of HTTP traffic, namely traffic seen at a caching-proxy server, we expect that though our invariants will overlap with the ones identified in [1], they will not be the same. Workload invariants for proxy-server traffic are listed in Table 2. These invariants are discussed in detail below.

Number	Name	Description
1	Median file size	$\approx 2K$
2	Mean File size	less than 27KB
3	File Types (accesses)	90%-98% of accessed files are of types Graphics, HTML, and cgi-map
4	File Types (bytes)	Most bytes accessed are of type graphics
5	% of accesses to unique Servers	less than 11%
6	% of servers referenced one time	less than 5%
7	Accesses concentration (servers)	25% are responsible for 80%-95%
8	Bytes concentration(servers)	25% are responsible for 90% of the bytes accessed
9	Success Rate	88%-99%
10	Self-similarity	$0.59 \leq H \leq 0.94$

Table 2: Invariants for Web Proxies

4.1 File sizes

Table 3 shows file size statistics for the workloads. For this analysis, we removed all files that have length zero, such as dynamically-generated or (cgi-bin) files. The median size in all workloads is very close to 2K; the minimum for the median is 1938 bytes

Workload	Max.	Mean	Median	1st Qu.	3rd Qu.	S
DEC1	22620000	8792	2087	386	6831	86758
DEC2	31690000	8598	1939	320	6582	88312
BU(G)	16710000	27480	2259	766	7028	336495
BU(U)	14160000	16240	2652	1046	6876	145939
Korea	56220000	14640	2222	506	7197	152902
VT-Lib	12780000	7008	2070	738	6044	68958
VT-CS	20440000	9762	2115	557	6225	123637
VT-Han	26860000	8272	2261	864	6064	95956
AUB	6779000	8863	1938	560	5615	96405
AOL	46290000	7545	2030	790	5963	127056

Table 3: File size statistics.

and the maximum is 2658 bytes. This appears in table 2 as an invariant. The mean file size ranges between 7K to 27K; this is consistent with the invariant for average server file size found in [1]. This is our second invariant and also appears in Table 2.

The value for third quantile is less than the average file size and the maximum file size is large when compared to the median and mean file sizes, this implies that the file size distribution is heavy tailed. The variability in file size is due to the different file types; video and audio file sizes are typically huge compared to text files. A detailed study to characterize file sizes with statistical distributions is in progress [9].

4.2 File types

Table 4 shows the percentage of accesses for each file type in each workload. Graphics Files are the most accessed type in all workloads. However HTML and graphics do not represent 90% of the total accesses which is different from the servers invariants in [1]. By comparing the results from the different workloads we notice that BU(G), BU(U) and Korea follow the results reported in [1]. However, in the other workloads HTML and graphics represent less than 89%. This is especially in the DEC data which has a *no type* category. Based on table 3, we can identify a third invariant namely that in all workloads, graphics is the most accessed file type followed by HTML followed by map/cgi.

The map/cgi percentage is lower in the BU workloads than in the other workloads. This could mean that the percentage of *dynamic* documents is going up over the years. Documents which are *dynamic* cannot be cached; hence if they have a higher percentage in the log file then we can predict that effectiveness of caching will decrease.

Table 5 reveals another invariant that is related to the previous one. File type Graphics is responsible for most of the accessed bytes. This invariant might not hold

File type	DEC1	DEC2	BU(G)	BU(U)	Korea	VT-Lib	VT-CS	VT-Han	AUB	AOL
graphics	59.1	58.5	82.8	84.8	66.8	67.1	51.3	69.5	67.5	73.9
html	11.3	10.9	12.8	12.5	21.7	15.0	20.6	14.8	13.6	14.2
map/cgi	1.8	1.6	2.1	1.5	5.7	12.7	19.2	8.7	9.3	8.2
audio	NA	NA	0.1	0.1	0.2	0.0	0.2	0.1	0.1	0.0
video	NA	NA	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0
compressed	NA	NA	0.2	0.0	0.4	0.0	0.1	0.0	0.0	0.0
ps/pdf	NA	NA	0.1	0.0	0.0	0.0	0.3	0.0	0.0	0.0
other	4.3	4.5	1.8	1.0	4.9	5.0	8.3	6.7	9.3	3.5
no type	23.5	24.5	NA	NA	NA	NA	NA	NA	NA	NA

Table 4: Percentage of accesses for each type

for a long time, since video and audio files are becoming more popular.

File type	DEC1	DEC2	BU(G)	BU(U)	Korea	VT-Lib	VT-CS	VT-Han	AUB	AOL
graphics	38.0	40.1	21.4	48.5	48.8	64.1	46.0	53.9	52.1	59.9
map/cgi	1.0	1.0	2.3	2.0	2.7	1.5	0.8	1.1	1.5	7.9
html	8.0	7.5	10.0	13.9	7.7	13.3	15.0	9.3	7.8	11.2
audio	NA	NA	28.7	6.5	2.1	1.7	5.2	7.5	0.1	1.5
video	NA	NA	3.5	18.5	12.5	11.6	8.0	13.7	27.4	5.0
compressed	NA	NA	6.6	3.8	14.4	1.5	6.8	2.8	1.0	2.2
ps/pdf	NA	NA	10.3	0.3	2.6	1.2	9.2	0.6	0.0	0.4
other	32.6	30.8	17.1	6.3	9.1	5.1	9.0	11.0	4.4	11.95
no type	20.4	20.5	NA	NA	NA	NA	NA	NA	NA	NA

Table 5: Percentage of bytes transferred for each type

By comparing Tables 4 and 5 we observe that types such as video, audio, compressed and PostScript/Portable Document Format (PS/PDF) constitute a very small percentage in number of accesses, however the percentage of transferred bytes for such types is significant. The files of type compressed include files that has the name extension *ZIP, GZIP, Z, etc.*. This implies that for groups of clients which access these types more frequently, caching should be *task* or *type dependent*, since this will maximize weighted hit rate [5].

Workload	%of accesses to unique servers	% of servers accessed one time	% of accesses to unique URLs	% of URLs accessed one time
DEC1	11.2	4.9	59.2	51.0
DEC2	11.3	4.9	60.1	51.8
BU(G)	1.3	0.2	1.8	0.5
BU(U)	0.5	0.0	0.9	0.2
Korea	1.3	0.3	2.8	1.2
VT-Lib	8.8	3.1	13.4	8.4
VT-CS	5.1	1.3	9.2	5.1
VT-Han	2.4	0.3	6.4	3.7
AUB	4.5	0.9	9.5	5.4
AOL	2.4	0.5	6.8	3.9

Table 6: Concentration of references to URLs and servers.

4.3 Concentration of references

Table 6 shows the percentage of accesses to unique servers, which is less than 12% in all workloads, so approximately 90% of the accesses are to the same set of servers. This is identified as an invariant and is listed in Table 2. The table also shows that the percentage of servers which were accessed only once is very small, less than 5%; this is another invariant shown in Table 2. On the other hand, percentage of accesses to unique URLs, and percentage of URLs accessed one time, had close percentages in all workloads except for the DEC workloads.

The small percentages of accesses to unique servers and URLs, and servers and URLs accessed one time, suggests that the locality of reference is very high for all workloads and that caching should be very helpful. To check this assumption we use a simulation and check the hit rate and the weighted hit rate for the workloads.

4.4 Distribution of accesses

In this section we examine the distribution for clients accesses and the distribution of accesses to servers and URLs. To do this we plot percentage of accesses vs. percentage of servers, URLS, and clients. Figure 1 shows that 25% of the servers get 80%-90% of the accesses while the other 10%-20% gets the rest of the accesses. This is another invariant that is listed in Table 2.

By examining Figure 2 we see that in all workloads, except the DEC workloads, 85%-95% of the accesses go to 25% of the paths. The workloads which share this behavior have similar graphs and access distributions. The figure also shows that accesses to URLs has a heavy tailed distribution.

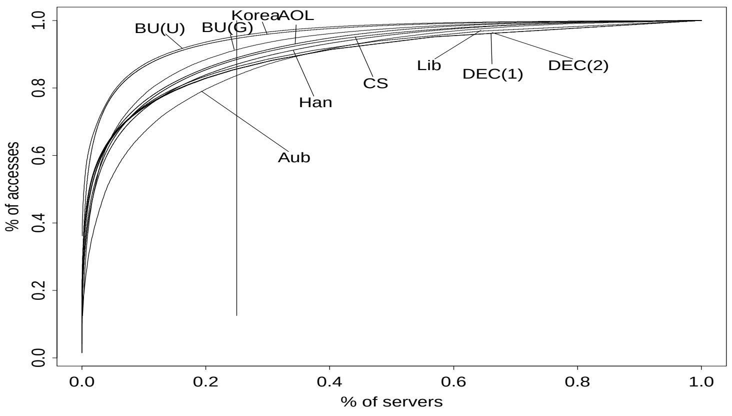

Figure 1: Servers concentration of accesses

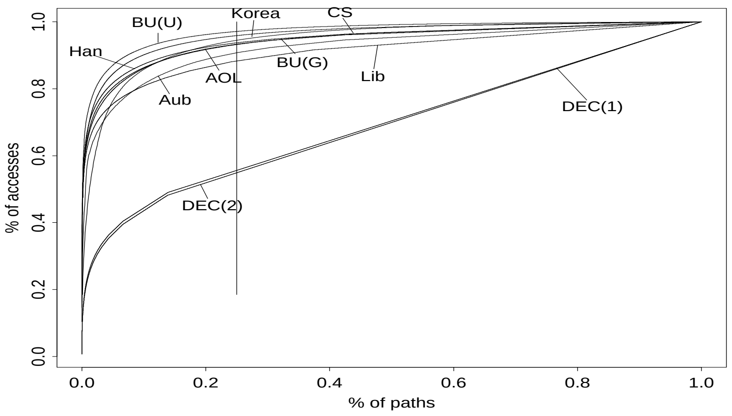

Figure 2: URLs concentration of accesses

We examined client access distributions to see if we can find invariants across workloads. The percentage of clients versus percentage of accesses made by those clients is plotted in Figure 3. Except for the Boston workloads, in all other workloads 50% of the clients are responsible for 80%-95% of the accesses. This could be true because of the existence of multiuser machines where users can login and run multiple instances of the network browser. Two cases that represent the extreme points in the graph, Computer Science at Virginia Tech (VT-CS), and Boston University BU(G) are collected from similar environments, i.e., from computer science departments and graduate students. However in reality the behavior is completely different. The BU(G) curve is almost linear and approximately 50% of the clients are responsible for 50% of the accesses. On the other hand, 50% of the clients in the VT-CS workload are responsible for more than 95% of the accesses. One reason for the difference is that the In VT-CS most of the accesses come from one multiuser machine. Other machines are accessed much less. In the BU(G) workload their are only five workstations, however in the VT-CS workload we have over thirty clients most of them are multiuser machines.

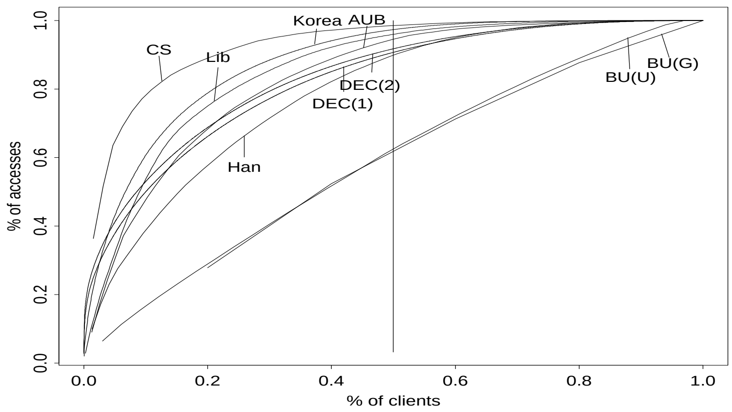

Figure 3: Clients concentration of accesses

Figure 4 shows that 90% of the bytes transferred come from 25% of the accessed servers. This is an invariant and is listed in Table 2.

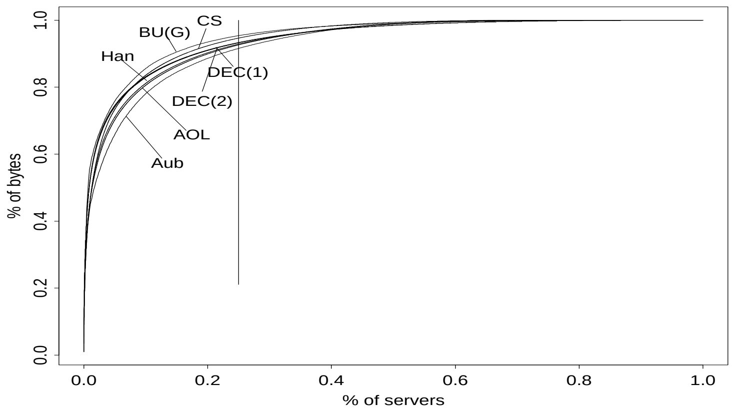

Figure 4: Servers concentration of bytes accessed

4.5 Rate of success and not modified files

Retrieving documents with no errors appears with high percentage in all workloads. We assume that a file is retrieved with no problems if it returns one of the following status codes in HTTP: 200 (*success*), 304 (*not modified*) or 204 (*OK but no contents*).

In all workloads we notice this percentage is in the range 88%-99%. We include this invariant in Table 2. The 304 return code is of particular interest for us since it reflects the percentage of files that are retrieved from the local or proxy cache.

Interestingly the status code 400 *client error (bad request)* is zero in all workloads except the DEC workload where the error appears 1.8% and 2% of the time.

Code	DEC1	DEC2	BU(G)	BU(U)	Korea	VT-Lib	VT-CS	VT-Han	AUB	AOL
200	72.9	71.1	NA	NA	84.4	84.5	86.3	87.2	85.1	NA
204	4.6	4.8	NA	NA	0.0	0.1	0.0	0.0	0.4	NA
302	1.7	1.8	NA	NA	1.0	3.2	2.2	1.9	4.2	NA
304	15.1	16.4	NA	NA	5.6	10.6	9.4	9.3	8.5	NA
400	1.8	2.0	NA	NA	0.0	0.0	0.0	0.0	0.0	NA
401	0.1	0.1	NA	NA	0.1	0.1	0.1	0.1	0.0	NA
404	1.1	1.1	NA	NA	1.7	1.1	1.5	1.1	1.2	NA
500	0.1	0.2	NA	NA	6.0	0.0	0.0	0.0	0.2	NA
other	2.5	2.3	NA	NA	1.0	0.2	0.4	0.2	0.4	NA

Table 7: Status codes

4.6 Self-similarity

Self-similarity can be defined informaly as, looking at the data with different time resolutions does not change the distribution. In other words, the data has similar properties to fractals where you can "zoom in" into the fractal and see the same structure.

This time independent structure of the data is shown in Figure 5. The figure shows the arrival times for a 17 day segment of a trace of all HTTP packets that have appeared on the Computer Science. The x axis represents time and the y axis represents number of HTTP packets appearing on the network. These graphs can be used to test for the existance of self-similarity in the data, however it does not prove that the data has this property.

The tests for self-similarity in the collected data showed that these workloads are self-similar. Table 8 shows the estimated *Hurst* or H parameter [10] for tested workloads. We have generated two time series for each workload. The first one is *access rate* and the second is *bytes rate*. *Access rate*, is defined as the number of accesses that a proxy gets per unit time. *Bytes rate* is defined as the number of bytes that a proxy sends per unit time. We used the R/S statistics and the normalized variance tests to estimate the H value [6]. For self-similar traffic H should be between 0.5 and 1. The closer H is to one, the stronger self-similarity in the tested traffic. The Boston traffic

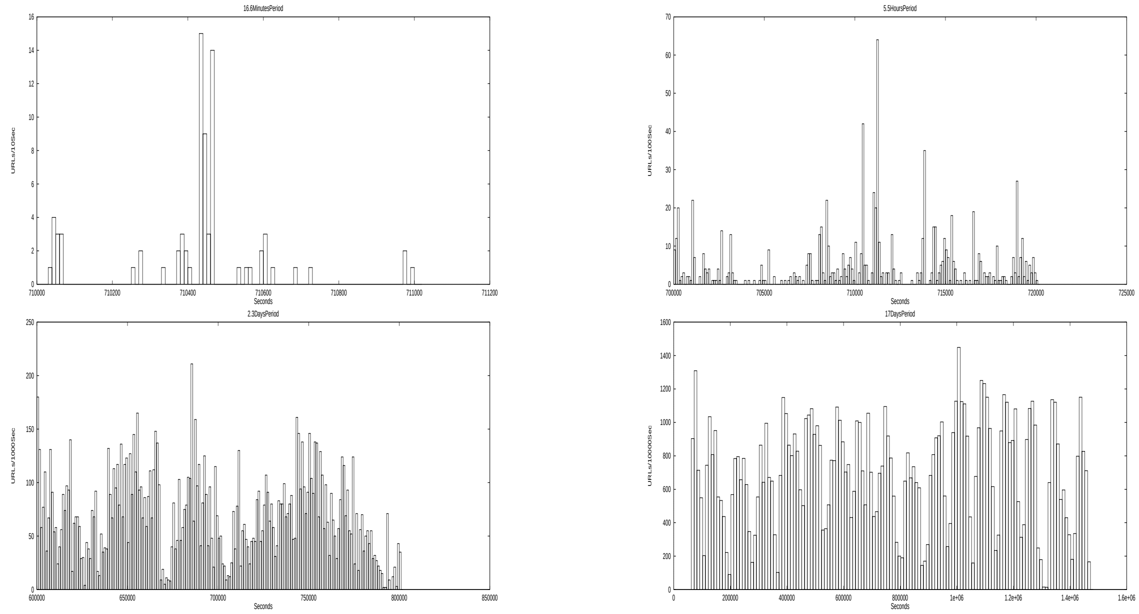

Figure 5: Visual test for similarity

	ACC/Min		Bytes/Min	
Workload	R/S	Var	R/S	Var
VT-Lib	0.93	0.93	0.81	0.86
VT-Han	0.79	0.75	0.59	0.65
AUB	0.94	0.85	0.82	0.81
VT-CS	0.86	0.82	0.80	0.70

Table 8: Values for the Hurst parameter

has been proven to be self-similar [6]. Self-similarity is an invariant and it is listed in Table 2.

4.7 Invariants

Table 2 shows a list of the identified invariants across the workloads. Thus we establish Hypothesis 1 and 2, given in section 2. Some of the identified invariants can be linked directly to caching and can be used to estimate if caching will help or not. Invariants 5, 6, 7 and 8 should be good predictors for cache performance. Future studies to identify statistical distributions for file sizes, access rate, bytes rate, and inter-arrival time is in progress [9]. The initial results reveal that we can model some of these distributions using Weibull distribution with $0.5 <= \alpha <= 0.8$. A Pareto distribution also can be used to model such distributions.

The previous discussion to identify invariants reveals two important questions that

Month	# of accesses	Bytes transferred(MB)
January	55916	333.12
February	66589	481.89
March	37473	239.02
April	59542	410.43
May	71217	433.65
June	11486	69.21
July	35936	214.48
August	40141	167.60
September	65395	356.05
October	100033	582.41
November	26657	203.86

Table 9: Accesses and bytes transferred for each month.

should be explored. First, do the identified invariants hold over time? Second, which of those invariants have more effect on caching? In the rest of this paper we will attempt to answer these two questions.

5 Longitudinal study of invariants and self similarity.

In this section we test if the previously identified invariants hold for the VT-CS workload over time. Thus we establish Hypothesis 3 in section 2. To do this we split the workload into monthly log files and applied the same analysis that we used to identify the invariants across workloads. We used the VT-CS log file because it has the longest duration since it spans over 11 months. Although the conclusions drawn from such an analysis cannot be generalized since we are using one workload, still we can identify trends or changes that appear over time.

Table 9 shows the statistics for each month for the VT-CS data. We notice that accesses during the summer are lower than accesses during the regular academic year. Number of accesses is not large compared to other workloads since most of the students accesses are to local servers which were filtered out.

5.1 File size statistics

Table 10 shows file size statistics for each month. Some of the previously identified invariants are still true for the monthly analysis. Mean file size is still in the same range; in addition the median is also around 2K. However we notice that the mean

Month	Max.	Mean	Median	1st Qu.	3rd Qu.	S
January	6876000	7806	1963	439	5749	82113
	20440000	9232	1976	509	6060	139921
March	3964000	8707	2060	431	6196	73368
April	14580000	9328	2015	468	5812	108827
May	12620000	9163	1884	407	5927	104259
June	2836000	11940	2608	849	7214	74181
July	16810000	10910	2250	768	6468	150187
August	5759000	8115	2176	758	6283	67139
September	17350000	10680	2250	687	6405	172139
October	13820000	11070	2371	780	6773	127460
November	16950000	13890	2544	838	6922	182887

Table 10: File size statistics each month

file size seems to grow with time. To test if this is true or not we performed a linear regression with the mean as the response variable and month as the dependent variable. The t statistic has a value of 3.05 and the slope is 391 which means that the model is a good predictor and that the mean file size is increasing over time. The increase appears to be due to the increase in the maximum file sizes available from Web servers. This increase will continue as long as the users can access such files over the network with minimum problems.

5.2 File types

Table 11 shows percentage of accesses to each file type for every month. Another invariant from the previous section holds across the months, that is graphics and HTML account for the the highest percentage of accesses. We test if the percentages will change over time. Linear regression revealed that the graphics file type *static* is decreasing by time and the map/cgi (*dynamic*) is increasing over time. Will this phenomena continue? How much does it affect caching? The answer depends on new technologies such as Java. In the future we may be able to reproduce images using programs which means that we will have more dynamic (un-cache-able) documents.

Table 12 shows percentage of bytes transferred for each type per month. Types such as video, audio, ps/pdf and compressed appear with significant percentage due to their big sizes.

5.3 Concentration of references

Table 13 shows that other invariants from the previous section, concentration of references (invariants 5,6,7 and 8) hold over time. However we test if it stays the same over

File type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
graphics	59.6	57.7	57.3	56.3	53.1	40.9	44.4	48.8	47.2	43.8	49.1
map/cgi	9.0	7.8	8.2	10.6	16.2	32.1	24.0	22.6	26.9	32.3	25.0
html	23.4	26.1	25.0	22.9	21.3	16.8	20.3	19.0	16.9	16.0	18.9
audio	0.2	0.1	0.0	0.1	0.0	0.1	0.3	0.2	0.1	0.2	0.2
video	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.1
compressed	0.0	0.0	0.1	0.1	0.2	0.1	0.0	0.2	0.2	0.2	0.0
ps/pdf	0.2	0.2	0.3	0.6	0.5	1.2	0.3	0.2	0.2	0.0	0.1
other	7.4	7.9	8.9	9.2	8.6	8.7	10.6	9.1	8.4	7.4	6.6

Table 11: Percentage of accesses for each file type

File type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
graphics	52.5	57.5	48.7	40.0	48.0	40.5	38.1	54.4	45.3	46.6	39.9
map/cgi	0.4	0.3	0.4	0.1	1.1	0.1	0.6	0.1	2.5	1.3	0.9
html	16.8	20.1	18.9	13.6	14.8	11.1	12.5	14.0	15.6	12.2	12.3
audio	0.8	5.7	5.2	3.5	1.4	4.4	20.9	1.7	2.3	10.1	6.2
video	13.3	5.8	1.4	6.4	11.4	2.0	0.8	7.2	7.2	7.6	21.2
compressed	2.5	4.1	1.0	3.5	3.9	6.3	3.9	5.0	17.9	10.0	5.9
ps/pdf	8.7	7.4	11.8	21.9	8.5	19.9	14.4	7.1	6.1	2.8	3.2
other	5.1	9.5	7.9	10.8	10.9	15.7	8.8	6.4	4.6	11.01	10.5

Table 12: Percentage of bytes transferred for each file type

time. A linear regression for each one of the variables with respect to month showed that the model is significant and that there is a positive relationship between each variable and the month. The t statistics is over 2.5 for all the variables. Hence, there is a strong evidence that those percentages are increasing which means that locality of reference is decreasing.

5.4 Rate of success and not modified files

The identified invariant for success rate in the previous section holds true over the year for the VT-CS data. From table 14 we notice that the percentage of the code 200 *(success)* is increasing. However the value of the the status code 304 *(not modified)* is decreasing. This means that fewer files are being accessed from the cache. This is consistent with what we have noticed with respect to percentage of accesses to unique servers and URLs and with the increase in map/cgi files.

Month	%of accesses to unique servers	% of servers accessed one time	% of accesses to unique URLs	% of URLs accessed one time
January	4.4	0.8	8.9	4.2
February	4.3	0.7	8.5	4.0
March	5.2	0.8	10.9	5.6
April	4.1	0.7	8.6	4.4
May	4.6	0.9	8.4	4.3
June	6.9	1.3	10.8	5.5
July	6.2	1.5	10.0	5.6
August	5.8	1.5	9.6	5.3
September	5.7	1.3	9.6	5.5
October	4.9	1.1	8.7	4.8
November	9.2	2.8	13.1	7.5

Table 13: Concentration of references to URLs and servers per month

5.5 Autocorrelation and self-similarity

One of the identified invariants for the workloads from the previous section the self-similarity. We have shown that VT-CS data is self-similar. The value obtained for the Hurst parameter from the R/S plot was 0.86 and from the normalized variance plot was 0.82. For $0.5 \leq H \leq 1$ the correlation decays to zero slowly so that

$$\sum_{k=-\infty}^{\infty} \rho(k) = \infty$$

and the process has long memory [10]. As a result we expect that the access rate for the VT-CS data will have a strong correlation. To test for this we extracted accesses/minute, accesses/hour, accesses/day, and accesses/week. We then generated a time series and plotted the autocorrelation function with different lags. Figure 6, which is the ACF plot for accesses/hour, shows that the correlation is strong and cyclic. The daily cycle is very clear in the figure; every 24 hours we get a peak.

In [6] the authors tried to identify sources of self-similarity in the Web traffic by showing that transmission times may be heavy-tailed, primarily due to the distribution of Web file sizes. The transmission times correspond to the *on* times and user think time corresponds to the *off* time. Multiplexing *on/off* times will generate a self-similar traffic.

Table 8 shows that self-similarity appeared in the tested data sets to be stronger with access rate than with bytes rate; this made us question the assumption in [6]. The file sizes might not be the major cause for self-similarity. After all the user may spend

Code	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
200	81.7	81.5	82.4	82.6	84.6	90.1	88.2	84.6	90.0	92.9	90.5
204	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
302	3.1	2.4	2.2	2.0	2.2	1.7	1.9	2.3	2.1	1.9	2.2
304	12.6	13.8	12.4	13.6	11.1	5.9	8.4	10.8	6.0	3.4	4.6
400	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
401	0.2	0.1	0.2	0.1	0.1	0.1	0.0	0.1	0.1	0.1	0.1
404	1.8	1.8	1.8	1.4	1.5	1.9	1.0	1.5	1.3	1.4	2.7
500	1.1	0.1	0.4	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
Other	0.5	0.3	0.6	0.3	0.3	0.3	0.4	0.6	0.4	0.3	0.3

Table 14: Status codes

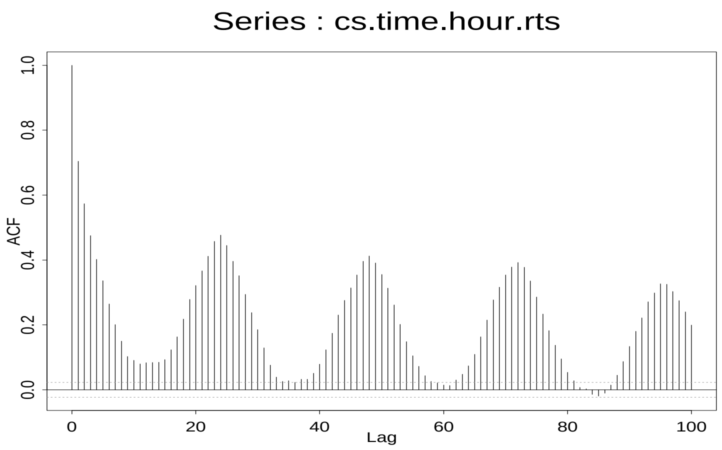

Figure 6: ACF for the VT-CS data by hour with Lag 100

15 minutes reading a text which is 5K in size and seconds looking at a picture which is 1MB. So file size and think time might not be related.

We think that another source of Web self-similarity is the repeated behavior of Web users since they do things in a repeated and correlated manner relative to minutes, hours, days and weeks. For example, in a school or a company environment, users will start accessing the Web slowly at the beginning of the day and in time more users will join the active ones. At lunch hour the number of accesses will go down and then build up again during the afternoon and the evening. Note that this phenomena is clear with a group of clients since most of the time they are in the same time zone. However servers get hits from all over the world and the cyclic or periodic behavior will be weaker. This could be the reason why self-similarity was not identified as one of the invariants for the server workload in [1].

The method we have used to examine the hypothesis in [6] is not formal, and our

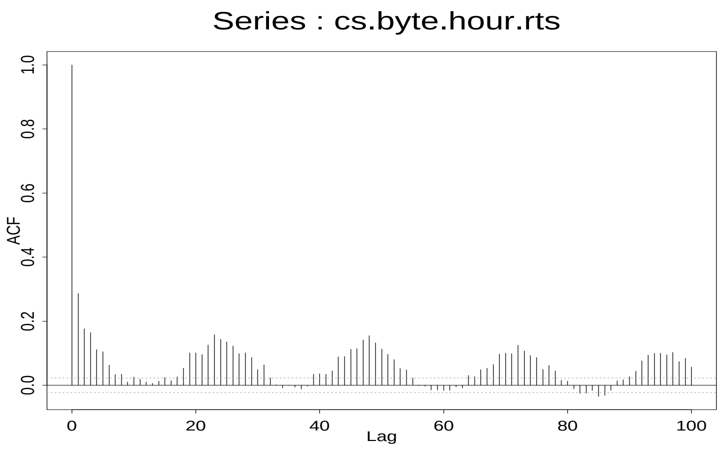

Figure 7: ACF plot for bytes/hour, VT-CS data

intention is not to falsify their hypothesis. We only try to show that there might be other sources for self-similarity and that it is user dependent. Figure 6 shows that there is a significant correlation between hourly access rate. This leads us to hypothesize that a major source for self-similarity is the way users interact with Web browsers. We also examined the ACF plot for accesses/minute, accesses/day and accesses/week. In all of them the correlation (ρ) was different from zero.

Since we are using accesses/hours then this behavior is the aggregate of several users' behavior over this hour. If file sizes are a source for Web self-similarity then including it in this graph should have an additive effect on the graph and we should see a higher correlation. Instead of accesses/hour we generated the time series bytes/hour for the same data set. Figure 7 shows that the correlation went down and it is decaying faster than the correlation for accesses/hour. This is supported by Table 8 which shows that self-similarity is weaker for bytes/rate. This implies that bytes transferred or file sizes have a negative effect on self-similarity.

Although the results are not conclusive, it is still interesting to try and justify why the bytes rate reduces self-similarity. One justification is that users don't think about file sizes when they use the Web and most of the time file size information is not known by users. The only time when users worry about the size is when they encounter a delay or a slow response from the server. In this case, transmission times can be machine or network dependent not only user dependent.

By this we establish hypothesis 4 in section 2.

	Final Size of Cache (MB)	Overall Hit Rate (%)	Overall Weighted Hit Rate (%)
DEC1	10,159.7	6.219	2.72
DEC2	9,881.9	5.470	2.74
Boston(G)	265.2	81.146	38.650
Boston(U)	507.9	85.489	46.69
Korea	10,078.3	63.107	51.224
VT-Lib	498.8	26.719	13.474
VT-CS	780.5	28.273	22.208
VT-Han	1,695.9	43.682	31.393
AUB	75.9	36.700	30.366
AOL	3,672.5	50.166	38.194

Table 15: Theoretical maximum hit rate and weighted hit rate

6 Invariants and Caching

Table 15 shows the Infinite cache hit rate for each workload using simulation. We ran all the workloads through a trace driven simulation of a proxy server with infinite cache. By examining Figures 1 and 2 we expect the DEC workloads to have the lowest hit rates and Table 15 confirms this result. Also by examining the same figures we expect that BU(U), BU(G) and Korea workloads to have the maximum hit rate which is also confirmed by the results in Table15.

7 Conclusions and Future Work

In this paper we identified a set of invariants that hold true across different proxy workloads and one of the identified invariants was self- similarity. We also showed that some of the identified invariants hold for the VT-CS workload overtime. One of our observations is that most of the accesses go to a small set of servers and URLs. In addition, in the majority of the workloads, a small set of clients are responsible for most of the accesses. One question that we will try to answer is what happens when we combine all workloads together and examine the distribution of accesses per server. Is the set of popular servers across all workloads the same? If true, then does this set change over time and can the change be characterized?

We explored the sources for self-similarity in the VT-CS data and our conclusion is that a major source for self-similarity can be attributed to the periodic behavior of Web users. The periodic behavior appeared at minute, hour, day and week intervals.

More work is needed to explore the sources for Web self-similarity and see if we can practically benefit from it. The fact that self-similar processes have a long memory, should allow forecasting or predicting future behavior based on the logged history.

There are other invariants across the workloads that need to be identified. A longitudinal study similar to the one presented in this paper but with more workloads is necessary to explore the behavior of those invariants over time.

8 Acknowledgments

Dan Aronson provided the America Online traces. Members of the VT NRG provided helpful comments on the manuscript. NSF NCR-9627922 partially supported this work.

References

[1] M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants. In *Proc. SIGMETRICS*, Philadelphia, PA, April 1996. ACM.

[2] H. Braun and K. Claffy. Web traffic characterization: An assessment of the impact of caching documents from NCSA's Web server. In *Proc. 2nd Int. WWW Conference*, Chicago, October 1994.

[3] T. Kwan, R. McGrath, and D. Reed. NCSA's World Wide Web server: Design and performance. *IEEE Computer*, 28(11):68–74, November 1995.

[4] M. Abrams, C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox. Caching proxies: Limitations and potentials. In *4th International World-wide Web Conference*, pages 119–133, Boston, December 1995. <URL: http://ei.cs.vt.edu/-˜succeed/WWW4/WWW4.html>.

[5] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox. Removal policies in network caches for world-wide web documents. In *ACM SIGCOMM'96 Conference*, pages 293–305, Stanford University, California, August 1996. <URL: http://ei.cs.vt.edu/˜succeed/96sigcomm/96sigcomm.html>.

[6] Mark E. Crovella and Azer Bestavros. Self-similarity in world wide web traffic evidence and possible causes. In *Proc. SIGMETRICS*, Philadelphia, PA, May 1996. ACM.

[7] Carlos R. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of www client-based traces. Technical Report BT-CS-95-010, Dept. of Comp. Sci., Boston Univ., 11 Cummington St, Boston, MA 02215, July 1995.

[8] M. Abrams and Stephen Williams. Complementing surveying and demographics with automated network monitoring. *World Wide Web*, 1(3):101–119, July 1996.

[9] G. Abdulla. *Characterizing and Modeling World Wide Web Objects.* PhD thesis, Computer Sci. Dept., Virginia Tech, 1997. Dissertation in Progress.

[10] Jan Beran. *Statistics for Long-Memory Process.* Chapman and Hall, NewYork, NY 10119, 1994.