Multimedia Traffic Analysis Using CHITRA95*

Marc Abrams Stephen Williams Ghaleb Abdulla Shashin Patel
Randy Ribler Edward A. Fox

{abrams,fox}@vt.edu, {williams,abdulla,spatel}@ei.cs.vt.edu

Virginia Polytechnic Institute and State University
Department of Computer Science
Blacksburg, VA 24061-0106

April 7, 1995

Abstract

We describe how to investigate collections of trace data representing network delivery of multimedia information with CHITRA95, a tool that allows a user to visualize, query, statistically analyze and test, transform, and model collections of trace data. CHITRA95 is applied to characterize World Wide Web (WWW) traffic from three workloads: students in a classroom of network-connected workstations, graduate students browsing the Web, undergraduates browsing educational and other materials, as well as traffic on a courseware repository server. We explore the inter-access time of files on a server (i.e., recency), the hit rate from a proxy server cache, and the distributions of file sizes and media types requested. The traffic study also yields statistics on the effectiveness of caching to improve transfer rates. In contrast to past WWW traffic studies, we analyze client as well as server traffic; we compare three workloads rather than drawing conclusions from one workload; and we analyze *tcpdump* logs to calculate the performance improvement in throughput that an end user sees due to caching.

Keywords: tools, networking and communication, workload modeling, education

1 Introduction

What is the nature of distributed multimedia network traffic – that is, traffic between clients and servers over a communication network? Are the traffic characteristics generated by different users at different times entirely unpredictable, or can one characterize multimedia traffic?

Using the tool described in this paper — CHITRA95— one can visualize, query, statistically analyze and test, transform, and model collections of traces of client or server requests and replies. The domain used as a case study in this paper to illustrate tool use is educational. In this domain, students use a workstation-equipped classroom for a course on multimedia and retrieve materials from the World Wide Web (WWW) during class. Outside of class, students retrieve course materials on demand from a department-wide server. The material includes articles, video, pictures, and other materials from a digital library that includes IEEE and ACM papers used in courses.

*This work was supported in part through the National Science Foundation to SUCCEED (Cooperative Agreement No. EID-9109853). SUCCEED is a coalition of eight schools and colleges working to enhance engineering education for the twenty-first century. Support was also provided by NSF CISE Institutional Infrastructure (Education) grant CDA-9312611.

1.1 Why Analyze Multimedia Network Traffic?

The analysis of multimedia network traffic can answer a variety of questions. These questions are addressed specifically for WWW traffic in §4.

A first class of questions addresses how users (clients) make use of multimedia material through a network. What mixes of media types and distributions of file sizes are accessed by users? What file size distributions arise for each media type? How many servers are accessed by clients and with what distribution? Do different groups (e.g., two sections of a college course delivered over a network) of users have similar usage characteristics?

A second class of questions addresses the requests that a particular server receives. What is the time between successive accesses to a file in a server (of interest to cache designers)? What is the distribution of file sizes and media types requested?

A third class of questions addresses how to best use mechanisms to improve network performance, such as caching and prefetching in a network. A variety of caching mechanisms have been proposed (e.g., [5, 7, 9, 13]). Pitkow and Recker, in proposing a cache policy, state, "Surprisingly, many of these [caching algorithms] lack an empirical or mathematical basis." Are there patterns of access that users commonly perform that can be exploited by algorithms that cache or even prefetch information? Are there patterns of user access that permit predictive prefetching? How does cache performance vary with cache size? What is the distribution of the number of times that a file is accessed each day (suggesting a "working set size" for a user in a cache)? What performance improvement do users experience with caching?

1.2 Related Work

The most immediate need for multimedia analysis tools is in the WWW due to its rapid growth and widespread use. Tools to analyze WWW trace files of *server* traffic include *wwwstat* [3], *gwstat* [8], and WEBVIZ [12].

The tool *wwwstat* can generate the following reports about a log from an NCSA httpd server: the number of requests the server receives and the number of bytes in the server response. The report can be by day, by hour, by client country, by client domain, or by Uniform Resource Locator (URL). *wwwstat* can also process queries about the server log, because it can limit its report to regular expressions containing certain hostnames, IP addresses, server response codes, dates, hours of the day, and URLs. Graphic presentation of *wwwstat* output in the form of histograms is provided by *gwstat*.

Our tool (CHITRA95) can produce essentially the same reports and histograms as *wwwstat* and *gwstat*. In addition, our tool also provides various time-dependent analyses of trace data, analyzes cache hit rates, uses *tcpdump*[1] traces [14, Ch. 21, App. A] to analyze URL response times, uses several visualizations rather than just histograms, provides transforms to reduce and aggregate the trace data, analyzes client traffic traces, analyzes ensembles of traces rather than just a single trace, performs statistical tests to determine if different traces are homogeneous, and generates workload models.

WEBVIZ views the files in the WWW as a database, and visualizes the structure as nodes in a graph with edges representing hypertext links. The frequency and recency of file accesses from an NCSA http log are indicated by color and line thickness in the graph. Queries are supported to restrict what nodes are shown (e.g., clients in a particular domain). CHITRA95 does not provide a view like this.

The case-study here has three distinctions from past studies. First, we not only examine the requests that one specific server receives from any client, but also the complement: the requests that any server receives from one specific client. Second, we compare the traffic characteristics of different groups of users (e.g., classroom users versus undergraduate or graduate lab users). Third, we also collect hit rates and use *tcpdump* to measure improvements in transfer rate due to caching.

[1] *tcpdump* allows a network host to record the headers of all network packets whose source or destination matches certain IP network address and port numbers.

1.3 Outline

This paper is organized as follows. §2 lists requirements for a trace analysis tool and discusses how CHITRA95 addresses these. A workload experiment is described in §3, and then questions from each question class in §1.1 are addressed for WWW traffic in §4.1 to §4.3, respectively. The key conclusions from analysis are in §5.

2 Requirements

2.1 Requirements for a Multimedia Network Traffic Analysis Tool

We believe that a tool to analyze network traffic traces should provide the following facilities.

Ease of use: It should be possible for a person to use the tool for five minutes and get interesting information about their trace files.

Ability to handle ensembles: A single trace represents the actions occurring for a single period of observation which may or may not be representative of other observation periods. User behavior can vary dramatically. If a trace analysis tool is to provide robust information to network designers that is likely to characterize user behavior over a variety of conditions, then the tool must analyze not one but a set, or *ensemble* of traces. The ability to handle ensembles is essential to being able to compare and contrast different types of workloads.

Query facility: The tool should answer queries about traces that have been collected so that qualitative questions about user behavior can be posed as queries. For example: Does the traffic on all Sundays during a semester look similar? How do the access patterns of students in dormitories differ from users of lab-based computers? Which servers on campus appear to be saturated in their ability to serve client requests?

Visualization: Visualization can summarize large amounts of trace data – for example, histograms of the sizes and types of files accesses. However visualization facilities in a tool should also represent time-dependent behavior in a system, such as the variation of requests to a server with time of day or day of the week.

Ability to perform statistical tests about ensembles: Common tests answer whether a set of traces is homogeneous, for example whether the traces are likely to have been drawn from the same distribution. This permits conclusions such as: Does one group of users with caching see a significantly different response time than a second group without caching? Do two user groups represent the same workload?

Ability to generate a workload model: Network designers require traffic models to use in analytic modeling and simulation. Thus the ability to fit a model to trace data is essential to generating a workload model.

Ability to detect common user access patterns: Common patterns could indicate the need to redesign the interface to a multimedia application to streamline the operations. Patterns could be exploited in data layout on server disks or when designing cache policies.

Ability to transform trace data: Transform methods can reduce the volume of trace data to speed analysis. For example, suitably chosen transforms can remove "noise" from a trace to reveal patterns or exceptions in traced behavior, or filter out accesses to certain types of media.

Tool extensibility: A tool should be user extensible in a language of the user's choice so that a user can codify commonly used analysis procedures to save time, or even write new analysis or visualization modules. There should be an easy way for tool users to share new modules in the form of a library.

2.2 How CHITRA95 Meets the Requirements

CHITRA95 is the third generation of a system to analyze trace data from computer and communication networks [1, 2]. CHITRA95 takes as input ensembles of trace files. A trace file is a sequence of $L > 0$ ordered pairs $(T_0, E_0), (T_1, E_1), \ldots, (T_{L-1}, E_{L-1})$ where $T_0, T_1, \ldots, T_{L-1}$ is an ascending sequence of numbers representing some index of time, such as timestamps, and $E_0, E_1, \ldots, E_{L-1}$ is a sequence of events. An event is an ordered tuple representing some actions of interest during observation of a system. For CERN or NCSA WWW server (http) log files, each T_i (for $0 \leq i < L$) represents a timestamp in the log, E_i represents a GET performed by a client, and the ordered tuple contains three components: client name, URL being requested, and file size.

CHITRA95 provides a library of at present 60 commands that each visualize, query, statistically analyze or test, transform, or model trace data. CHITRA95 is operating system and graphical user interface (GUI) independent. At present there is an automatically generated X-windows interface to the toolkit to provided an integrated analysis tool. A forms interface to permit remote use through the WWW is being implemented.

Ease of use: To satisfy ease of use, CHITRA95 contains a mega-command that invokes many interesting and commonly used analysis and visualization commands on a collection of log files, which each may be either a CERN or NCSA server log or a *tcpdump* log. The mega-command generates a collection of visualizations and summary statistics on a monitor or writes a file in one of a variety of formats (e.g., PostScript, GIF) and tables of summary statistics.

For each *tcpdump* log, the mega-command produces the following analysis of file transfers: statistics, such as mean, min, max, standard deviation of file transfer rate; histogram of transfer rate; transfer rate by source or source-class; transfer rate by destination or source-class; and graph of server traffic load versus time. For each CERN httpd [10] trace, including a trace from httpd configured to run as a proxy [9], the mega-command produces several results. These include statistics, such as total number of requests (e.g., URL GETs); histogram of number of requests per client or client-class histogram of number of file type; histogram of file size; histogram of request destination, showing the number of requests that go to each server or server-class; a time dependent graph of the rate of cache hits and misses; and visualization of time since last access to a file as a scatter plot versus time and as a histogram over all time.

We are currently implementing a World Wide Web forms interface to CHITRA95, so that a user need not even take the time to install a copy of CHITRA95 on their machine. The user can electronically mail trace data to a WWW server running CHITRA95, then open a URL that contains a form to select the types of analysis desired, and then view the results of analysis as a set of dynamically-created Web pages containing visualizations and tabular data. With the user's permission, the traces and created pages remain on the server, so that the server acts as an archive of the traffic characteristics of various sites that submit traces. This would facilitate studies that characterize traffic from many sites, rather than from a single site. If the user finds the initial results useful, then they can invest more time to install and learn how to use the full set of CHITRA95 commands to do more detailed analyses.

Ability to handle ensembles: CHITRA95 allows an ensemble of traces to be operated on as a unit, so that the same analysis, transform, and modeling can be performed on all traces in the ensemble. Some visualization methods combine the data from all traces in an ensemble into one graph; others graph data from each trace separately. We routinely handle ensembles with thousands of short traces.

Query facility; ability to detect common user access patterns; ability to transform trace data: CHITRA95 provides each of these through a set of *queries* that retrieves data from trace files matching a query criterion and *transforms* that apply a function to map the matched data to a new form. For example, there is a query to identify all patterns in an ensemble of trace data, and a transform to replace a pattern by an aggregate representation of the data to simplify the trace data. Pattern matching can identify common user access patterns. There is a query to identify the most often occurring components in trace data, and a transform to eliminate all but these components from the trace data for further analysis. There are also transforms to remove all but certain components in the trace data vector, to perform arithmetic transforms such as scaling the range of vector component values, and to aggregate vector component values (useful to group server ids into categories).

Visualization: The visualization methods in CHITRA95 include two dimensional plots, such as scatter plots and Gannt charts, histograms, periodograms, and correlograms. The tool is unique in its ability to visualize and model *categorical* time series data. (Categorical data has no total ordering among its values; examples include hostnames, domain names, and URLs. In contrast, numerical data has a total ordering; examples include the sets of integer and real numbers.) Categorical data — but not as a time series — arises in *wwwstate*, and the histograms for categorical data (e.g., client domain, client sub-domain, and URL) are sorted by the number of requests so that the histogram bars are non-decreasing. CHITRA95 provides novel visualizations to represent the time evolution of categorical data.

Ability to perform statistical tests about ensembles: The primary test in CHITRA95 is the Kruskall-Wallis (KW) rank sum [11, pp. 422-425], which tests the hypothesis that two or more samples are drawn from the same distribution. The implementation includes the correction for the case when the samples include many ties. The KW test is used because it makes no assumption about the underlying distribution of the data. The KW test is used also in another test for stationarity (as defined for stochastic processes) of trace data, to identify whether one segment of a trace file "looks like" another segment.

Ability to generate a workload model: CHITRA95 can generate one of three types of workload models: a semi-Markov process, a model based on the Chi-square Automatic Interaction Detection (CHAID) procedure [6] to find events in traces that are likely to occur in combination with other events, and a novel time-dependent stochastic process.

Tool extensibility: CHITRA95 is a toolkit that consists of a set of small programs that communicate through a standard, self-describing file format. The programs form a library to which a user can add new programs. Therefore a user can extend the toolkit by writing a script that combines existing modules to codify analysis procedures, or by writing a new program to add a command to the CHITRA95 library. In either case the user can use any language of their choice.

3 Example: WWW and Three Educational Workloads

We use as a case study traces from World Wide Web clients representing three classes of educational traffic at Virginia Tech. The Computer Science Department provides materials for thirteen courses through the WWW (accessible from http://ei.cs.vt.edu/courses.html). Four of these classes are "paperless" — all course material is delivered during class using the Web, students retrieve all assignments from the Web, and in one class take exams and quizes through the Web. In several other courses the Web is used for lecture delivery during class. In all classes the Web serves as a repository for items such as lecture notes, assignments, course syllabi, and links to departmental documents (e.g., honor code description, computer use handbooks). Students retrieve the course material either from networked workstations during class; from campus computer labs; from dormitories with SLIP or Ethernet connections; and, through the Blacksburg Electronic Village, from campus apartments or homes through Ethernet or SLIP over 14.4kbit/sec modems.

The course use of the Web at Virginia Tech is targeted to exploit multimedia, primarily through a project to create a digital library that allows students to obtain on-line copies of papers used in class. [4].

Other multimedia types of class material include scanned images of diagrams, photographs, and links to servers outside of the Virginia Tech campus.

If the use of network delivery grows in popularity on college campuses, then a significant amount of campus network and even Internet related traffic will be education-related. Therefore in this paper we use CHITRA95 to report on preliminary analysis of four classes of educational workloads at Virginia Tech: (1) classroom access by students that each use a network-connected computer during a class on Multimedia; (2) undergraduate access by students in an undergraduate computer lab; (3) graduate access by students in the graduate computer lab; and (4) accesses to the main server for educational materials (host *ei.cs.vt.edu*).

The following equipment is used. In the classroom workload (1), each client is Netscape running on a 10baseT Ethernet-connected Apple PowerPC 6100/60 AV, and the proxy server runs on a thin-net connected DECstation 3000. In the graduate (2) and undergraduate (3) lab workloads, the clients are Netscape and Mosaic and run on DECstation 3000s or 5000s, the proxy server runs on a DECstation 5000, and all machines are connected by thin-net Ethernet. In the server workload (4), the server is a DECstation 3000 connected to thin-net. The 10baseT and thin-net networks are inter-connected by an FDDI network that is then connected to the Internet by a T1 link (soon to be upgraded).

Study objectives: The objective of our study is to characterize and compare the four workloads, to identify the performance improvement possible by caching off-campus URL gets, to characterize the mean time between accesses to the same URL, and to qualitatively characterize the predictability of URL gets.

Factors: We vary two factors during this study: the size of cache used on a proxy server, and the type of workload (i.e., (1) to (4) described above).

Experimental method: Studies in the literature and statistics available on the Internet report on server traces, rather than client traces. That's because it is easy to collect a server trace: just turn tracing on at a server. The resultant log says which client used one specific server.

However it is hard to collect traces of *client* behavior that record how users access *any* network server. Either one must collect a trace at every host supporting a Web browser client or one must ask all clients to use a certain proxy server and then record a log on the proxy server. We chose the later route. On multi-user student machines, we created modified versions of the Mosaic and Netscape commands that caused clients to use one of several proxy servers so that we could record client behavior.

To conduct the experiment, we first installed CERN proxy servers to cache off-campus URLs, and we enabled logging. Therefore the proxy server log traces all URL accesses by clients that have their proxy server set to our proxy server, whether they are actually cached or not.

For the classroom workload, we used two proxy servers. The undergraduate and graduate workloads were each assigned to a unique proxy server. The workstations in the classroom were divided into two groups of 12 machines, and each group was assigned to one server. (The division was done in a way to avoid a bias, such as machines in the front of the room in one group, because there might be a correlation of student participation in class with the seat location.) The use of two groups allows comparison of different cache sizes for similar workloads – students in the same class at the same time.

We also installed *tcpdump* on each proxy server machine to log all packet traffic to or from the server. We varied the cache sizes (using randomization) each day. The data observation period for our study collects a complete log of all URL accesses by each client for every day in the Spring 1995 semester. In this paper, we report on a portion of the Spring semester. Finally, we analyzed the collected traces with CHITRA95.

Limitation in our experimental design: One problem with the CERN proxy server is that it cannot automatically invalidate outdated files in the cache unless an expiration date in the file is set. In addition, there is no way for Mosaic and Netscape clients to force a cache reload. Therefore, to avoid user complaints about outdated file copies, we configured proxy servers to force file expiration after 24 hours.

Furthermore, we cached only off-campus accesses, because students that are accessing course-related Web pages require the latest copy of a page. In addition, caching on-campus accesses would have questionable benefit, given that the time for a client to read the cache is not much different than the time to reach the true server on-campus.

4 Sample Analysis Session with CHITRA95

4.1 Client Traffic Analysis

First consider the distribution of file sizes. The histogram in Fig. 1 represents one day of all client requests in the undergraduate lab workload. The histogram suggests an exponential distribution, with the vast majority of files transferred being less than 1kB. The shape of the graph and the 1kB figure is generally representative of all three workloads. However, occasionally another distribution is superimposed upon the general distribution, as shown in Fig. 2; the long bar at 17000 bytes is due to a single URL that was accessed many times repeatedly within a single day. This illustration comes from the undergraduate workload.

Analysis of media type shows different distributions for the different workloads. For both undergraduate and graduate browsing, the distribution of media types manifested similar trends. First, the number of GIF files far outnumbered other file types (Fig. 3). Second, the number of requests for GIF files was approximately twice the number for HTML files. This fact, along with the file size information, leads us to believe that the majority of HTML files accessed in a browsing environment have inline graphic images. A histogram of the sizes of GIF files for one day of the graduate workload is shown in Fig. 4; the histogram for other days and for days in the undergraduate workload appear similar. The histogram shows that the vast majority of GIF images are under 500 bytes, probably representing buttons or icons. We conjectured that the multimedia class would access HTML files with more GIF files than did the other workloads, but measurement showed the opposite trend (which turns out to relate to instructor authoring habits). In the majority of days of the multimedia class, the number of HTML file requests was approximately twice the number of GIF requests (Fig. 5); however many more audio files (labeled AU) were requested than in the other workloads.

Finally, the distribution of servers accessed by the multimedia class for one day is shown in Fig. 6. The distribution suggested by the histogram is representative of other days in the multimedia class. The graduate and undergraduate lab workloads tended to have long histogram bars, showing that a small number of servers were accessed much more than on average.

4.2 Server Traffic Analysis

In this section we analyze the NCSA httpd log from the main server machine used for educational course materials in Computer Science at Virginia Tech, named ei.cs.vt.edu. The log file contains data for the period of 11-January through 27-March 1995. A total of 229,256 files were accessed by 2522 different clients during that time period. Plotting filesize distribution (not shown) indicates the same exponential distribution as the client logs with most of the files under 1000 bytes. The file type distribution is very near to that of the multimedia class workload with respect to the proportion of HTML files to others and the proportion of HTML to GIF files (about twice as many HTML files as GIF files).

The log file is visualized in Figure 7 using a periodogram unique to CHITRA95 for categorical time series data. The x-axis represents time; 11 January is day 0 and 27 March is near the right end of the axis. Each graph point represents a client request. The y-axis value of a point is the recency, or *time since the file requested was last referenced.* Moving across the graph along the x-axis, there are vertical bands of white representing the early morning period of each day when no one was accessing the server. The completely white vertical band near day 65 corresponds to a period when the server went down for installation of a peripheral. The density of graph points is greater below the line $y = 0.1$ hours than above the line. Also there is a second density decline between 10 and 100 hours. These breakpoints are indicated by Fig. 8, which shows the data from Fig. 7 as a histogram showing the distribution of

Date	*No cache*	*10Mbyte cache*	*50Mbyte cache*	*Homogeneity hypothesis test*
1 March	2403.05 ($N = 340, F = 3090$)	N/A	2729.92 ($N = 451, F = 4305$)	accept
8 March	1608.38 ($N = 665, F = 4223$)	1867.10 ($N = 290, F = 6514$)	N/A	accept
21 March	1439.29 ($N = 850, F = 4935$)	N/A	1595.77 ($N = 321, F = 5599$)	accept
22 March	2108.16 ($N = 785, F = 6178$)	N/A	1673.51 ($N = 893, 7659$)	reject

Table 1: Transfer rates, in bytes/second for multimedia class workload. N denotes sample size; F denotes mean file size transferred. Two hypotheses are tested: that file sizes requested by the two halves of the class are drawn from the same distribution, and that the resultant transfer rates are drawn from the same distribution. Kruskall-Wallis test with $\alpha = 5\%$ and correction for ties is used for hypothesis tests.

last-access time through the eleven week period. The first histogram bar is significantly higher than the rest, showing that inter-access time of a file is most often on the order of tens of minutes. The height of the histogram bars then slowly declines until about 24 hours on the x-axis. After this, between 24 and 28 hours, there is an *increase* in the number of accesses, perhaps because students may work at the same time each day, and thus are likely to reaccess the same URL after 24 hours elapse. This suggests that a cache policy for this server should remove files after about either 30 minutes or 28 hours.

4.3 Effectiveness of Caching

Our conjecture before measurement was that caching client requests to off-campus servers should produce a larger performance improvement for the multimedia class, where students tend to request the same files for their workstations, than for the undergraduate and graduate browsing workloads, where client requests would not be likely to follow any patterns.

The method used for data collection was to analyze *tcpdump* logs recording packets received by and sent from ports assigned to the proxy servers for each workload. A CHITRA95 module first converts *tcpdump* logs into a sequence of transfer rates. Another CHITRA95 module is then used to compute basic statistics, such as mean transfer rate. Table 1 represents four days of multimedia classes. Each row corresponds to one day. Because the workstations in a class are partitioned into two halves assigned to different proxy servers, two transfer rates (with different cache sizes) in the table correspond to each day. The transfer rates are higher with caching for three days and lower for the fourth day. On three of the days, the homogeneity hypothesis that the file sizes of URLs requests by the two halves of the class is accepted (using CHITRA95's KW test), giving evidence that on these days the two halves of the class generated the same workload with respect to file size. On the other hand, on these three days the hypothesis that the transfer rates for the halves of the class that did and did not use caching was accepted, giving evidence that there is no improvement in performance due to caching. The only significant difference is on the day when transfer rate *decreased* with caching, but this occurs on the fourth day when the hypothesis that the workload is the same is also rejected.

Comparison of transfer rates with no cache and with a 50Mbyte cache was also performed on the graduate workload. The sample mean of transfer rate was 6288 bytes/second with no caching ($N = 8541$) and 6300 bytes/second with caching ($N = 12,883$). The KW test ($\alpha = 5\%$) indicated that the difference in transfer rates is not significant. The mean file size of a URL response was 4122 bytes when no caching was used, and 5376 bytes when caching was used. Because all clients use the same cache on each day, we expected that the workload for days with and without caching were not homogeneous; applying the KW test ($\alpha = 5\%$) on file sizes of URLs requested rejected the hypothesis was rejected.

The transfer rates in the classroom workload are lower than those of the graduate workload because in the classroom the client machines are slower than in labs and the network connection is through bridges. This was confirmed by performing an FTP of a 1Mbyte file from a classroom PowerPC to the server, yielding a transfer rate of 0.123 Mbytes/second, and from the graduate client host (a DECstation 3000) to the same server, yielding 1.00 Mbytes/second.

A primary goal of analyses run on these workloads was to determine the effectiveness of caching WWW pages fetched from outside the university. Visualization of cache hit rate was done on both a cumulative basis and a time basis. It was discovered that the rate of cache hits remains fairly constant throughout the trace (Fig. 9) and therefore the hit-ratio remains constant as well. For the graduate browsing workload the average hit-rate was approximately 17.4%. The classroom workload had an average hit-rate of 34.6%. This indicates that a classroom environment is more likely to benefit from the use of a cache server since many URLs will be accessed by most or all of the students in the class in a short period of time.

5 Conclusions

The CHITRA95-based WWW analysis tool is available from http://info.cs.vt.edu/~chitra/www.html. CHITRA95 can analyze any type of trace data, even though we discuss its use only for WWW traces. Our WWW traffic study yielded the following conclusions:

- Based on server traffic to the educational WWW server in our department, a cache strategy for this server should remove files after about either 30 minutes or 28 hours, based on a decrease in the distribution of inter-access times of files after 30 minutes and an increase between 24 and 28 days (because students are likely to work at the same time of day on successive days).

- We did not get the 50% improvement with caching server accesses that was noted by Glassman [5]. In our experiments, we were caching client off-campus accesses at a proxy server; thus the proxy server was a cache for the off-campus portion of the Internet. We expected a lower hit rate, and indeed observed this: the average hit rate for the multimedia classroom workload is about 34%, whereas the maximum hit rate for the graduate student workload is only about 17%.

- We had expected the cache hit rate for the proxy server to vary dramatically with the workload. Surprisingly, our time-dependent analysis (using CHITRA95's plot of rate of occurrence of hits and misses as a function of time) shows that the cache hit rate is remarkably constant through the observation period. The only exception is that for a brief period in one workload (graduate students) on one day (8 March), there was an interval when the hit rate was significantly higher, but then returned to the previous rate. This time dependent characterization would not have been possible without the rate visualization for categorical time series in CHITRA95.

- The distribution of file size in all but two days of the graduate workload suggests a Poisson arrival process. However in two days (27 February and 9 March) there is superimposed onto this arrival process a concentration of particular file sizes. This concentration is due to repeated access to a certain URL. The concentration of accesses on 9 March is to a URL which serves as the top level page for a tree of Web pages. We conjecture that the repeated accesses arise because the user or users are clicking on a "back" button that appears *inside* the Web page, which the Mosaic and Netscape browsers treat as a fetch to a new URL, rather than redisplaying the copy of a page that they had in memory earlier. The repeated URL serves as the "menu page" of the tree of Web pages, and thus is one to which users repeatedly turn. In contrast, had the user clicked on the "back" button that is part of the browser itself, then the URL would not have been refetched. It is thus interesting that a user interface feature — which "back" button a user prefers using — can create a "URL avalanche" of requests to a server, dramatically increasing server and even network load!

- Analysis of file-type accesses in the graduate and undergraduate lab workloads reveals that there are usually about twice as many GIF as HTML files fetched each day, and the GIF files are almost always under 500 bytes. Assuming that some of those GIF images are standalone, then there is an average of just under two embedded GIFs per HTML page in the laboratory workloads.

We plan to derive workload models from the entire suite of spring semester traces. Few such models exist in the literature, despite their potential value in succinctly characterizing traffic for multimedia network designers.

Acknowledgements: Alan Batongbacal wrote much of CHITRA95, with help from Anup Mathur and David Connerley. Carl Harris and Laurie Zirkle helped set up proxy servers to collect trace data.

References

[1] M. Abrams, N. Doraswamy, and A. Mathur. Chitra: Visual analysis of parallel and distributed programs in the time, event, and frequency domain. *IEEE Trans. on Parallel and Distributed Systems*, 3(6):672–685, Nov. 1992.

[2] M. Abrams, T. Lee, H. Cadiz, and K. Ganugapati. Beyond software performance visualization. *to appear in Concurrency – Practice and Experience*, Feb. 1994. Also appeared as TR 94-07, Computer Science Dept., VPI&SU.

[3] R. Fielding. wwwstat home page. <URL: http://www.ics.uci.edu/WebSoft/wwwstat/>, July 1994. Dept. of Information and Computer Science, Univ. of Calif. at Irvine.

[4] E. A. Fox and N. D. Barnette. Improving education through a computer science digital library with three types of WWW servers. In *Second World Wide Web Conference '94: Mosaic and the Web*, 1994. <URL: http://ei.cs.vt.edu/papers/WWW94.html>.

[5] D. Glassman. A caching relay for the World-Wide Web. In *Proc. of the First International World Wide Web Conf.*, Amsterdam, 1994. Elsevier.

[6] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.

[7] H. P. Katseff and B. S. Robinson. Predictive prefetch in the Nemesis multimedia information system. In *Proc. Multimedia '94*, pages 201–209, San Francisco, Oct. 1994. ACM.

[8] Q. Long. *qwstat* home page. <URL: http://dis.cs.umass.edu/stats/gwstat.html>. Univ. of Mass.

[9] A. Luotenen and K. Altis. World-Wide Web proxies. In *First International Conference on the World-Wide Web*, CERN, Geneva (Switzerland), May 1994. <URL: http://www.w3.org/hypertext/WWW/Proxies/ >.

[10] A. Luotonen and T. Berners-Lee. CERN httpd 3.0 guide. <URL: http://www.w3.org/hypertext/WWW/Daemon/User/Guide.ps>, Oct. 1994. CERN.

[11] L. Ott. *An Introduction to Statistical Methods and Data Analysis.* PWS-Kent, Boston, 3rd edition, 1988.

[12] J. Pitkow and K. Bharat. WEBVIZ: A tool for World Wide Web access log visualization. In *Proc. of the First International World Wide Web Conference*, Amsterdam, 1994. Elsevier. <URL: http://www.elsevier.nl/cgi-bin/WWW94link/31/overview >.

[13] J. E. Pitkow and M. M. Recker. A simple yet robust caching algorithm based on dynamic access patterns. <URL: ftp:ftp.cc.gatech.edu/pub/gvu/www/pitkow/prediction_analyzer/docs/caching_paper.ps.z>. Graphics, Visualization, and Usability Center, Georgia Tech.

[14] W. R. Stevens. *TCP/IP Illustrated, Volume I: The Protocols.* Addison Wesley, Reading, MA, 1994.

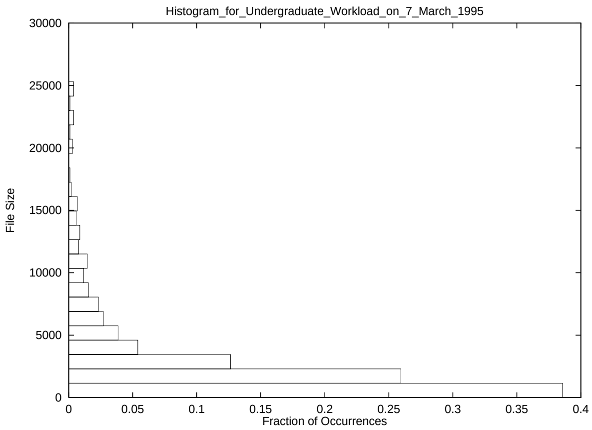

Figure 1: Typical file size distribution.

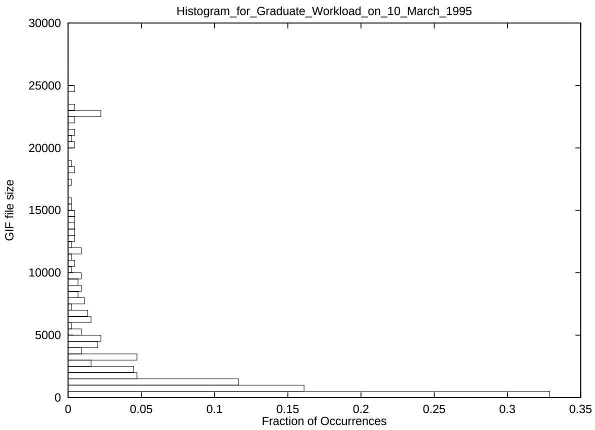

Figure 4: GIF file size distribution.

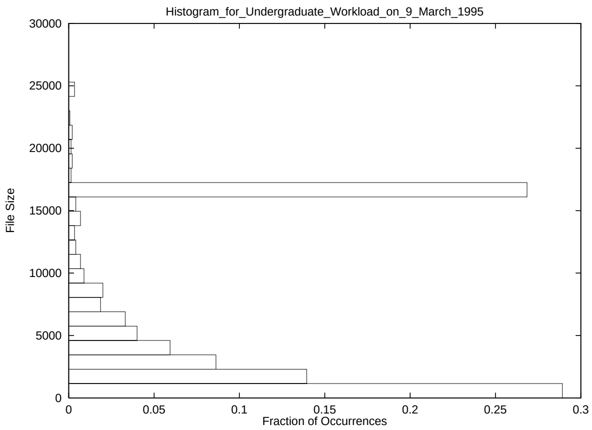

Figure 2: Outlier for file size distribution.

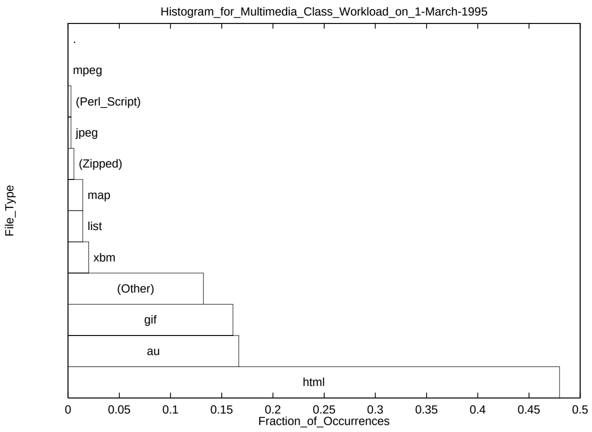

Figure 5: Media type distribution.

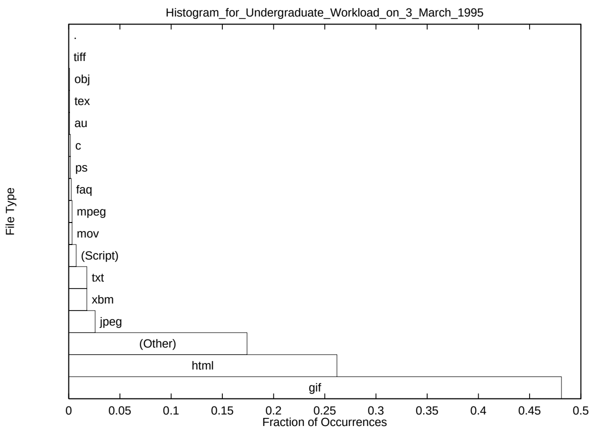

Figure 3: Media type distribution.

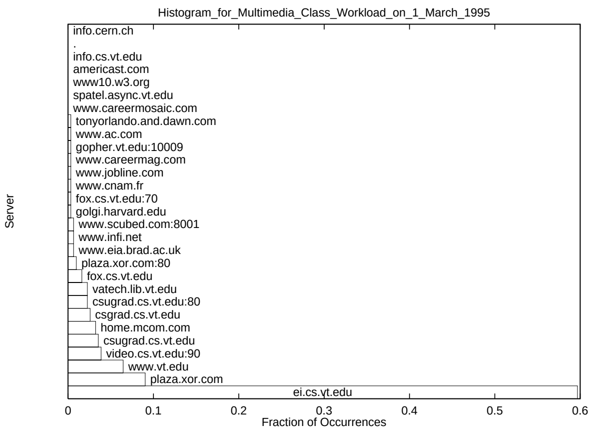

Figure 6: Distribution of servers accessed.

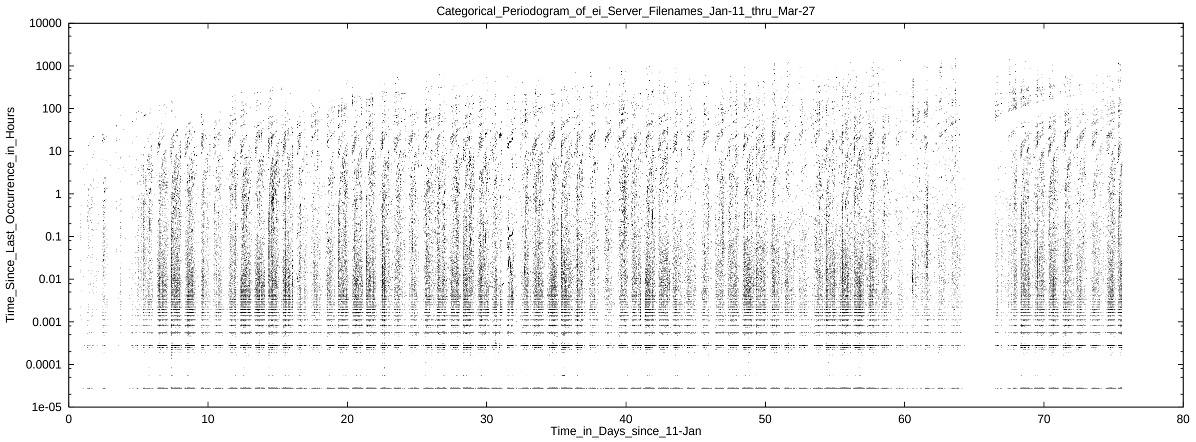

Figure 7: Periodogram of server accesses.

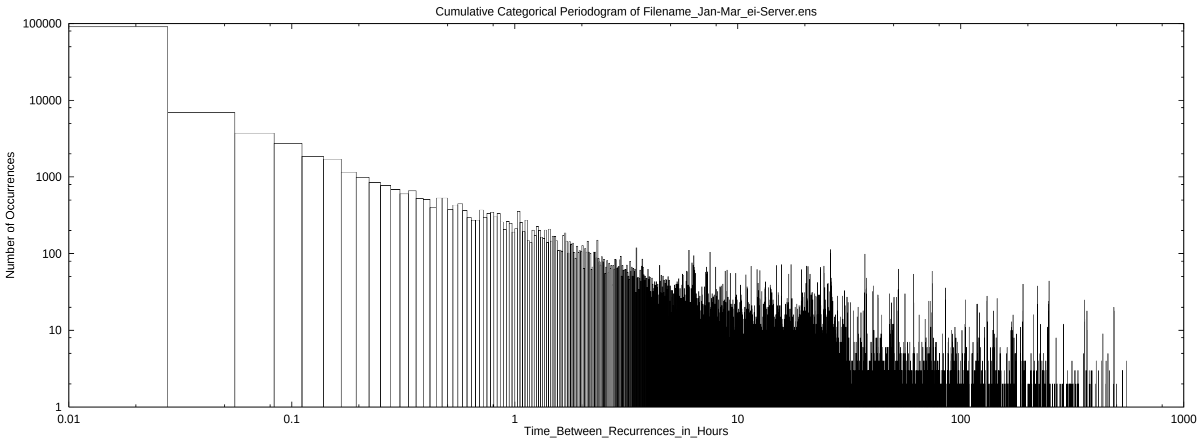

Figure 8: Distribution of last-access times

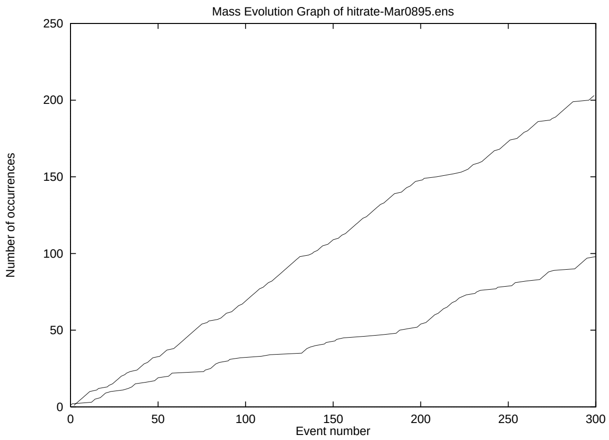

Figure 9: Time dependent hit and miss rates.