Palantir: A Visualization Tool for the World Wide Web

Nektarios Papadakakis Evangelos P. Markatos Athanasios E. Papathanasiou[1]
Institute of Computer Science (ICS)
Foundation for Research & Technology - Hellas (FORTH), Crete
P.O.Box 1385
Heraklion, Crete, GR-711-10 GREECE
markatos@ics.forth.gr

Abstract:

World Wide Web traffic increases at impressive rates reaching up to a several million hits (requests/clients) per day for busy Web servers. To serve all these clients effectively, it is necessary to have a good knowledge of their geographic distribution and access patterns. Understanding the geographic distribution of an organization's Web clients is essential in making important decisions that will reach the client base more effectively. For example, replication, caching, and advertisement have been widely used to improve information dissemination. However, these methods will be productive, only if made at strategic places on the Web, places that are *close* to the client base.

In this paper we present the design and implementation of Palantir: a tool that animates world wide web traffic. The tool displays the origin and magnitude of a Web server's hits either in real-time or in batch mode. It can synthesize the traffic to several Web server's so as to get a global view of the hits in a multi-site organization. Using Palantir, a user can get a deep understanding of *where* a server's clients are located, and thus, how to reach them more effectively.

Table of Contents

1. Introduction

World-Wide Web traffic continues to increase at impressive rates. Busy web servers may get as many as several millions of hits (accesses) in a day. Accesses may originate from all over the world and may result in a ‘‘rush hour” that lasts 24-hours per day. Web traffic will probably continue to increase as more people gain access and new applications (including commercial ones) are emerging.

To meet the demands of this ever-increasing traffic, webmasters should design their web servers in such a

---

[1] The authors are also with the University of Crete.

way as to disseminate information (and sell or advertise products) effectively and reliably. If a web server appears too sluggish, clients may easily seek the information they need elsewhere, because competition on the Web is just a mouse-click away. A first step towards effective information dissemination is understanding a web server's client base, and reaching out to it.

In this paper, we describe Palantir[2]. Palantir is a visualization tool that can be used to display the origin, volume and type of the incoming requests of a web server. It can display either a summary of the traffic over a given period, or an *animation* of the requests. It can be used to show in pictorial form the clients of a web server, as well as the number of requests and the type of files retrieved. The tool takes as input the log files of one or more web servers, and animates the amount and source of URL requests. The animation is overlayed on top of a geographical map, so as to show which continent, country and city the request was originated from. A frame of such animation is shown in figure 5. It shows the requests (originating from the Central and Eastern United States) made to the Web server of the Computer Science Department of the University of Crete: www.csd.uch.gr. The figure shows the type (color-coded) and the volume of requests. In addition to that, it suggests that the majority of requests originates from New York and Boston.

Palantir can be used to help people visualize the traffic requests and the client base of a given site. Once a webmaster understands the client base of a web server, he can use this knowledge in order to reach the client base more effectively. Such information about the client base can be useful in several strategic decisions that will have to be made by an organization. For example, suppose that a US-based busy web search engine would like to launch a mirror server in Europe. Where is the most promising location to place the new server? Studying the access patterns directed to the original web server which originate from Europe and surrounding areas will help the webmaster make an informed recommendation on the best location for the new server. Potential candidates would be those places in Europe that have a large number of clients, or *repeated* clients, or (repeated) *customers* (in case there is some product sold on-line by the server). Visualizing the client base will allow the webmaster understand which place in Europe will most probably maximize the profit of a new web server.

As another example, consider a virtual store that needs to be listed into several virtual malls. Listing the store in a mall involves some expenses (e.g. rent), but can also result in profit when purchases are made through the mall's customers. Deciding which are the most appropriate malls to list a virtual store into, is a cost-benefit analysis, which takes into account the number of shoppers and ''window-shoppers'' in each mall. If a virtual mall has a significant amount of sales, it is worthwhile to keep the virtual store there. Another mall could have a few sales, but several window-shoppers. In this case, it may be worthwhile to keep the store there, as well, in anticipation of increased sales.

As a final example, consider an organization whose web server experiences increased traffic during some periods of time, e.g. every Monday morning. To amortize such traffic bursts, the organization may rent a web server for these periods and redirect a portion of its incoming requests to this rented server [6]. The rented server contains a copy of all the information of the original server and can serve all incoming requests transparently. However, choosing the appropriate server to rent is a complicated decision, which must take into account the geographic distribution of an organization's incoming requests. In order to be effective, the rented server should be located close to the source(s) of traffic.

We believe that Palantir is a useful tool in visualizing and understanding the client base of a web server.

[2] The name of the tool is borrowed from *Palantiri*, the seeing stones mentioned in [5]. The Palantiri were stones connected to each other forming a web. Anyone looking into one stone could *see* what was going on in the other stones.

This understanding is valuable in making several crucial decisions that relate to reaching the client base in the most effective way. The rest of the paper is structured as follows: Section 2 presents the high-level design and interface of Palantir . Section 3 presents the structure and implementation of Palantir . Section 4 presents related work, and section 5 summarizes the paper.

2. Design - User Interface

The purpose of Palantir is to display the web traffic in a pictorial form and *lead* the user into a better understanding of the traffic patterns and their implications. The tool is completely written in Java to enhance its portability across different platforms.

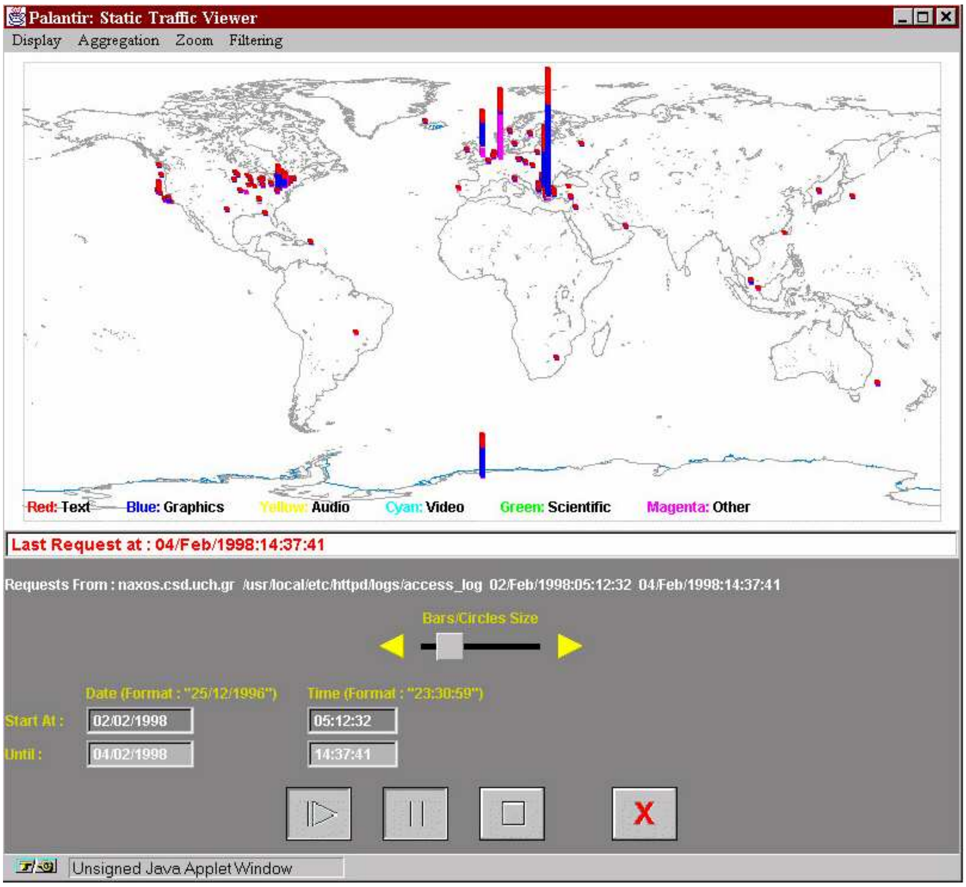
Figure 1: Control Panel for WWW traffic visualization

Figure 2: Configuration Window for WWW traffic visualization

The tool is started by directing a web browser, like Netscape, to a particular web server (currently `naxos.csd.uch.gr:9000`). After the connection has been established, a screen, like the one shown in figure 1, appears within the browser. In the center of the window lies Palantir 's main control panel, which provides two basic functions: the configuration of the servers that are going to be used and the choice of the mode, in which the visualization of the log files will take place (*static* or *dynamic mode*).

Configuration: Through the configuration window (*Configure Servers*), the trace(s) that are going to be studied may be selected. Once the configuration button is pressed, a window, like the one shown in figure 2, appears on the screen. The user can select to visualize one or more log files from one or more web servers. Each trace file can be located on any computer, connected to the Web. The only requirement is that a server (*Log Server*), that is able to read and manipulate the log files, runs on the specified computers. Animating the log files from several web servers is particularly useful in multi-site organizations, or in organizations that run several web servers (e.g. one for each research group or department), and would like to get an idea of the *total* traffic towards the organization.

Static and Dynamic Mode: Palantir can animate the web traffic in static mode or in dynamic mode. In the first case, the requests, which have occurred during a specific period of time and are contained in the selected log files, are animated in the viewer. Each request remains displayed until the end of the simulation (it has an unlimited time life). Thus, the stacked bars (or the concentric circles) present the total amount of requests cumulatively (summary of traffic over a specified period). In the Dynamic mode, Palantir 's viewer tries to capture the instant traffic of requests. Each request, contained in the log file, is considered to have a limited time life. As time passes, new requests are displayed on the viewer, while those that have exceeded their time life (old requests) are deleted.



United States, and Europe (Greece, Germany and the United Kingdom).

Palantir 's Traffic Viewers: By pressing the button labeled *Palantir View*, one of Palantir 's *Traffic Viewers* is displayed. Palantir supports two viewers: the *Static Traffic Viewer* (figure 3), and the *Dynamic Traffic Viewer* (figure 4). The upper part of both viewers is dominated by a map. Initially, the map of the whole wide world is displayed. However, the user may zoom in (or out) at the appropriate level of interest. For example, in figure 5 the user has zoomed in the Central and Eastern United States, while in figure 6 he has zoomed in the Mediteranean Sea.

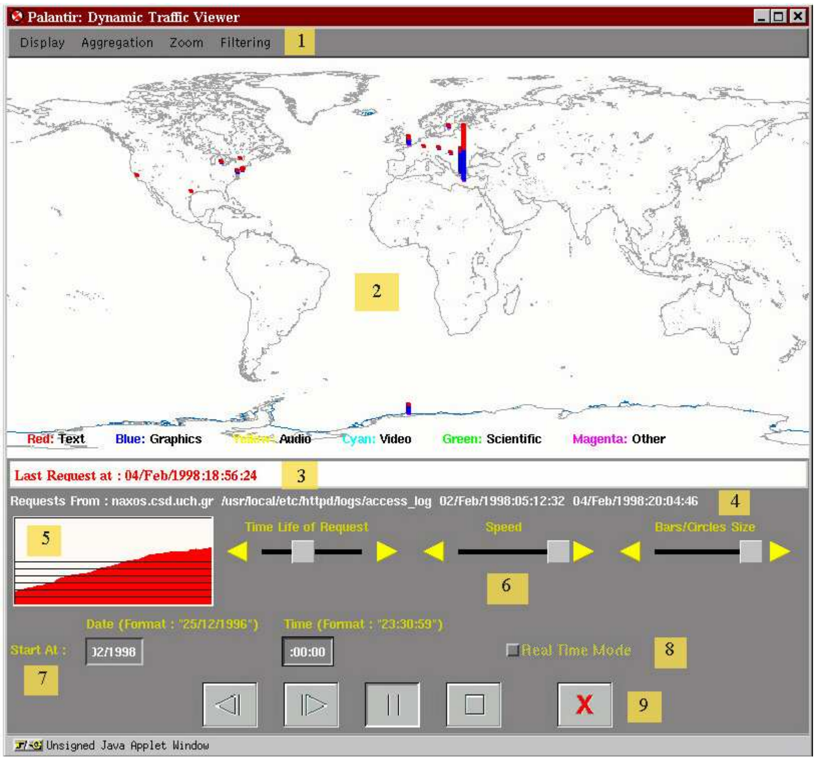


Figure 4: Palantir 's Dynamic Traffic Viewer Window. The majority of requests for the current time originates from Greece. The display also shows the diagram of the incoming requests, or equivalently the incoming load (in red). The Dynamic Traffic Viewer includes the following components: (1) the *Display*, *Aggregation*, *Zoom and Filtering Menus*, (2) the *Geographic map*, on which the simulation takes place, (3) the *Last Request field*, (4) the *Request From field*, (5) the *Loader*, (6) the *Control Scrollbars*, (7) the *Start At field*, (8) the *Real-time Mode button* and (9) the *Simulation Control Buttons*.

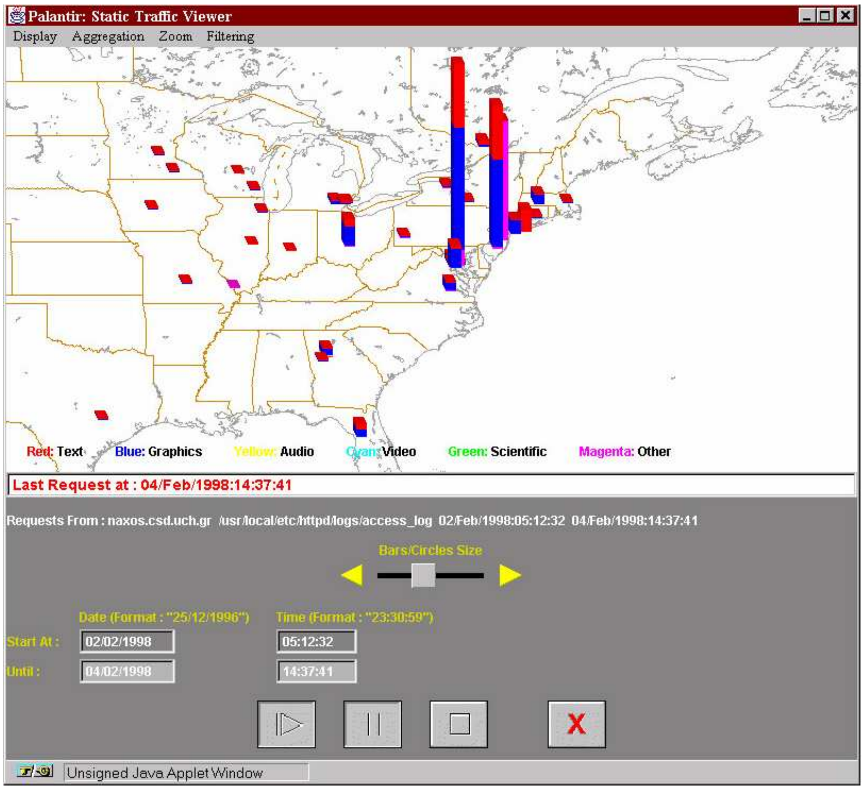


Figure 5: The Static Traffic Viewer Window after zooming in the Eastern United States.

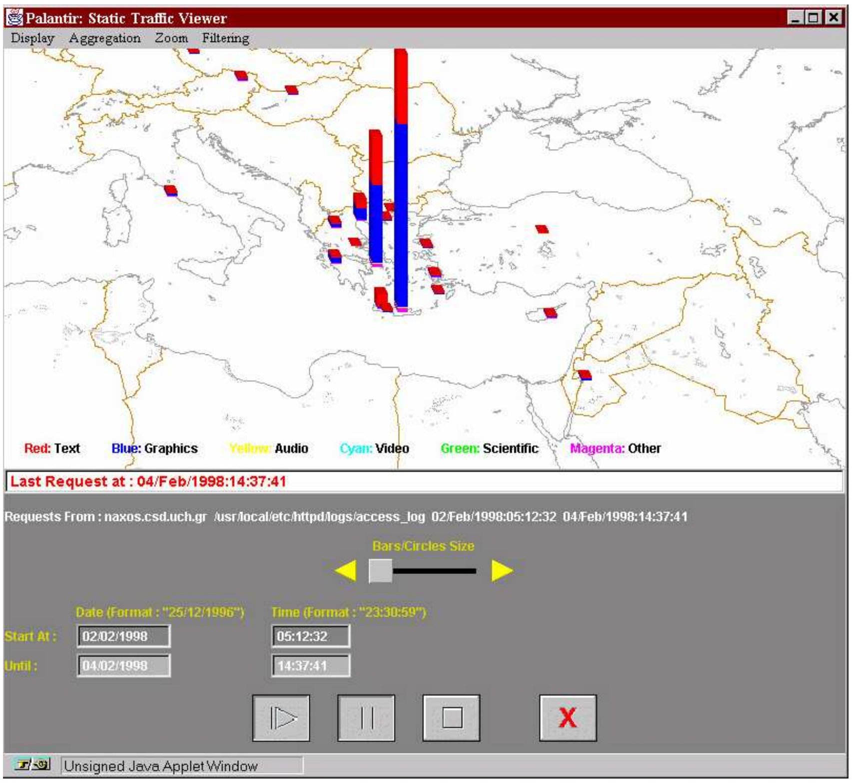


Figure 6: The Static Traffic Viewer Window after zooming in Mediteranean Sea. Several requests originate from the islands of Crete and Rodos, as well as the mainland of Greece.

The visualization of the log files is controlled through four menus located in the upper left corner of the Traffic Viewer Window (figure 4-1). From left to right, these menus are:

Display menu: The type and magnitude of requests that originate from each region are shown in the map either as *stacked bars* or as *concentric circles*. Concentric circles are useful to pinpoint clients with few requests, while stacked bars are more useful to visualize the traffic of very busy servers, since they effectively use a third dimension in data visualization (the height of the bar). Each bar contains several colors that represent the types of the files requested. Text files are presented in red, image files are in blue, audio files are in yellow, video files are in cyan, and other files are in magenta.

Aggregation menu: Palantir has the ability to aggregate the requests that originate from a broad geographic region to a single stacked bar (or concentric circle), displayed in the center of the specified region. Three types of Aggregation may be used:

- *No Aggregation:* Each request is displayed in its exact point of origin.
- *Default Aggregation:* Requests are aggregated from several predefined regions.
- *Custom Aggregation:* The user defines the borders of several regions, which are going to be used for aggregating requests.

Aggregation is useful, when the user wants to find out the total amount of requests, that comes from a very broad region, like a country or continent.

Zoom menu: Through this menu, the user is able to zoom in a specific location in order to study more effectively the traffic that originates from a particular geographic region. The ''Zoom menu'' offers nine different zooming levels that allow the user to zoom in the map as much as necessary. In addition to the above, by clicking on the world map the image zooms in the point, that was specified with the mouse. Two examples of zooming in are given in figures 5 and 6.

Filtering menu: The ''Filtering menu'' gives the ability to *filter* the animated requests. Palantir provides two kinds of filters: Domain Filter and Request Filter. The Domain Filter checks the domain name for a specified string. Only those requests that come from a domain, whose name contains the specified string, are displayed. Similarly, the Request Filter checks the name of the requested file. If it contains the specified strings, the request is displayed. The filtering is currently done via simple text-matching: the user supplies a text-mask, and the tool animates only requests that match this mask. In figure 7, the Domain Filter Window is presented. In this example, the visualization will focus on requests originating from educational nodes (containing in `.edu`).

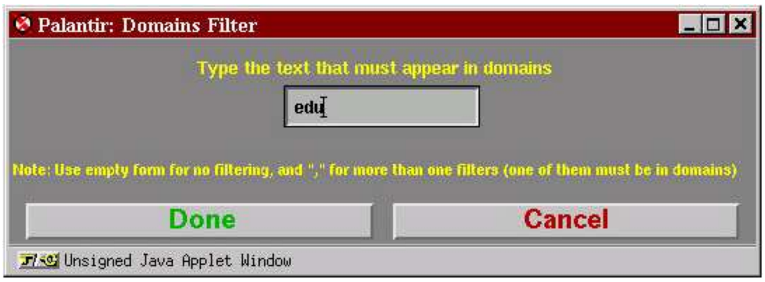


Figure 7: The Domain Filter Window: Only requests originating from educational nodes (containing `.edu`) will be simulated.

The second half of the traffic viewer contains information about the simulation and several control buttons (figure 4).

Last Request at field (3): In the ''Last Request at" field the timestamp of the request currently being animated is presented.

Requests From field (4): This field gives information about the log files being animated. Specifically, it displays the name of the log server or servers, the full pathnames of the log files in use and the timestamps of the first and last entry.

Loader (5): The loader animates the incoming load of requests. It is available only in the Dynamic Traffic Viewer.

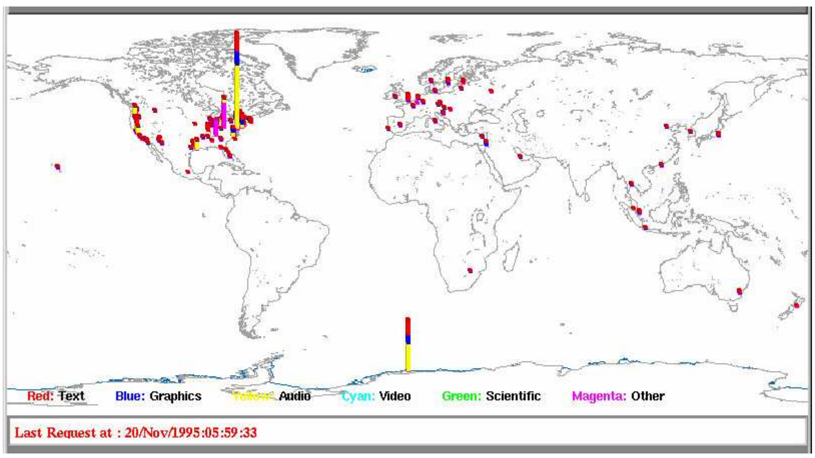
Scrollbars (6): The Dynamic Traffic Viewer contains three scrollbars that control the time life of each request (a hundred per cent will present cumulative results), the speed of the simulation and the size of the stacked bars (or circles). The Static Traffic Viewer has only one scrollbar, that controls the size of the stacked bars (or concentric circles).

Start At and Until fields (7): In the case that old requests, recorded in a log file are simulated, the Start At field of the viewers may be used to start the visualization from a specific timestamp. The default value is the timestamp of the first entry of the log file. The Until field can be used to indicate the timestamp at which the visualization of the log files should stop. The default value is the timestamp of the last entry of the log files being stimulated. The Until field is available only in the Static Traffic Viewer.

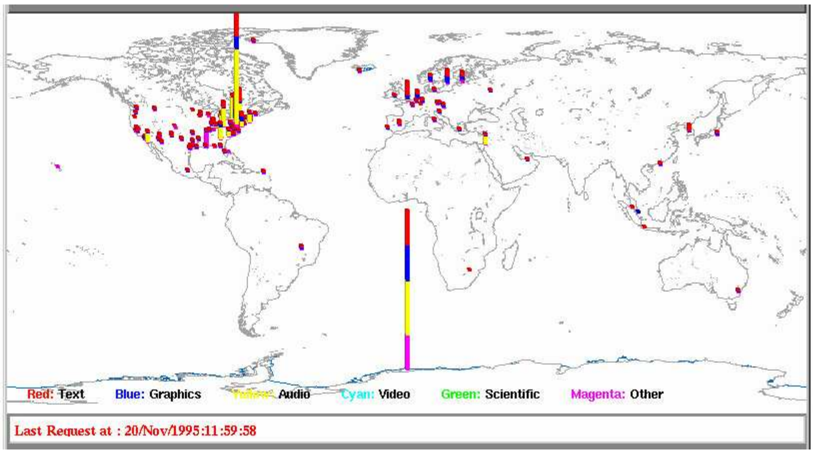
Real-time Mode (8): This button enables the viewer to operate in real-time (only new incoming requests are displayed). When the end of the log file is reached, the real-time mode is automatically enabled. Real-time mode is available only in the Dynamic Traffic Viewer.

Control buttons (9): In the lowest portion of the traffic viewer, there are several control buttons. Starting form left to right, the first button starts the simulation in reverse order (from newer to older entries). When the beginning of the log file is reached the simulation continues in normal order. This button is available only in the Dynamic Traffic Viewer. The second button starts the simulation in the normal order (from the beginning to the end). The third one pauses the simulation, while the fourth one stops it and resets the viewer. Finally the last button closes the traffic viewer window.

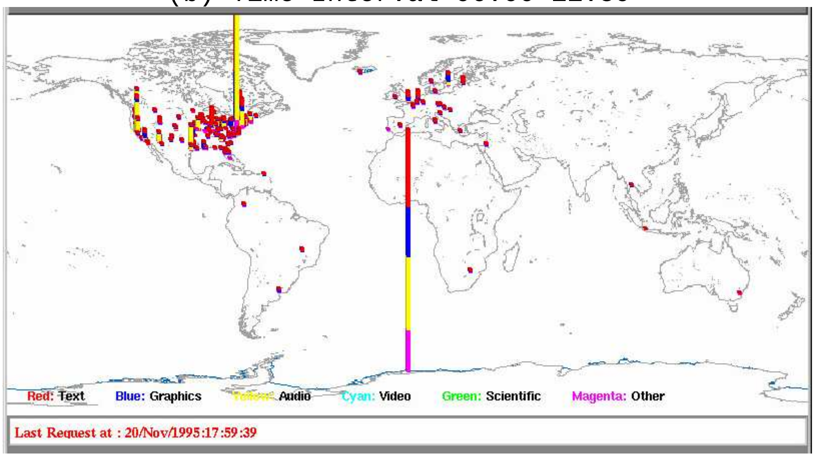
As a final example, figure 8 represents the requests accepted by the Web server of the University of Rochester during four different time intervals of the 20th November 1995. It is apparent that the Web server is busy during all day long, exhibiting a whole day rush hour. The incoming load is especially heavy during the time interval 06:00-23:59. During this period, the majority of requests comes from Antarctic and the Eastern United States.



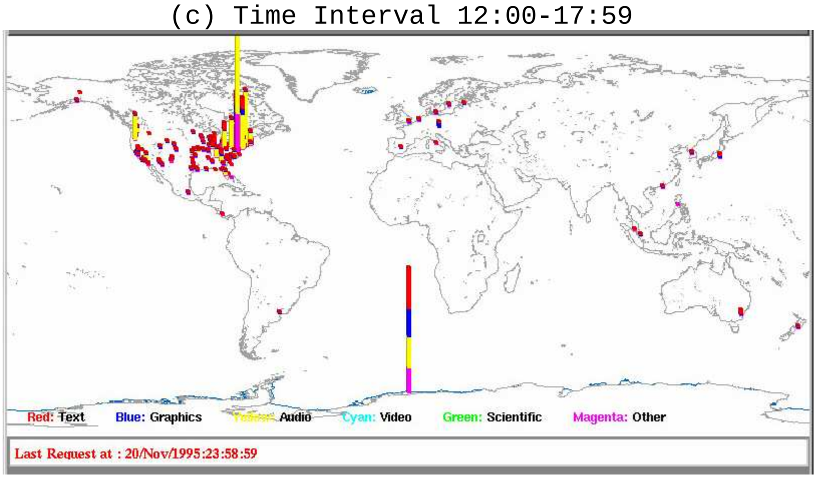
(a) Time Interval 00:00-05:59



(b) Time Interval 06:00-11:59



(c) Time Interval 12:00-17:59



(d) Time Interval 18:00-23:59

Figure 8: A whole day rush hour. Summary of the incoming traffic to the web server of the University of Rochester during four different time intervals of the 20th November 1995.

3. Implementation

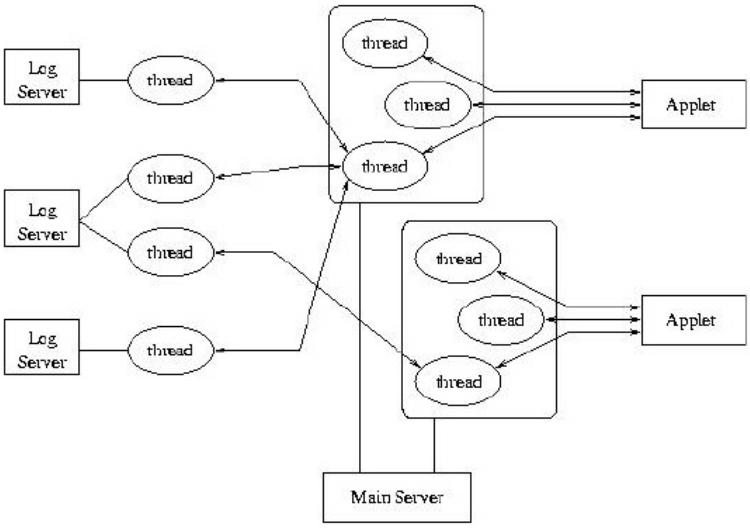


Figure 9: Structure of the Palantir visualization tool

The structure of our tool is shown in figure 9. It consists of three major components:

- **Log Servers:** The purpose of a log server is to read the log file.
- **Main Server:** The main server manages the geographical database and performs various chores on behalf of the applets.
- **Applets:** These implement the user interface.

All the components of the tool are written in Java.

The Log Server

The Log Server is an application program, whose task is to read log files and send them (via TCP/IP) to the Main Server. When is starts executing, it opens a socket on a given port and waits for requests to this port. When it receives a request, it spawns a new thread, which handles all further communication. Thus, a Log Server can serve concurrently several different Main Servers.

The Main Server

The Main Server is the most significant part of our tool. Its main function is to communicate with the applets that display the visual information on the user's web browser. It consists of three threads that perform its main functions concurrently. The first thread communicates with the log servers requesting the traces to be displayed. The second thread gives the geographic maps to the Applet. Each time the user zooms in the screen, a new map is needed to display the new data. These maps are downloaded from Xerox on-line map server at (`http://mapweb.parc.xerox.com`). To complement the above service, the tool keeps a local cache of the most frequently used maps. This cache helps in speeding up accesses to maps that were recently used, and to ensure the continuous operation of the tool in case the map server becomes unreachable: if the user requests a map, and the map server fails to respond, then a ‘‘similar” map is loaded from the local cache. The third thread deals with the translation of host names and IP addresses into its exact latitude and longitude. Unfortunately, this task is rather difficult. To our knowledge there is no standard method that can translate a host name located anywhere on the earth into latitude and longitude. Thus, we used the following mechanisms to help us in this translation:

- **Host Names:** in several cases the host name reveals the country the host is located. For example, host names ending in `.edu` are usually located in US, host names ending in `.uk` are located in the United Kingdom, etc.

- **Whois databases:** there exist several databases that give us detailed information for the location of domain names. Such databases can be found in `nic.ddn.mil`, and in `whois.internic.net`.

- **Manual data collection:** for some of the countries, we found lists with host names and the cities where they are located. For some of them we manually translated these cities into geographical coordinates.

Thus, the actual translation from host name (or IP address) to geographic coordinates is done as follows: For each IP address, we find its corresponding host name by a DNS-lookup procedure. To speedup this lookup, we keep a local cache with the associations between IP addresses and host names. Subsequently, for each host name we derive its domain. Based on the domain name, we find the country the domain belongs to (either by looking at the suffix of the domain name, or by querying a ‘‘whois” database). After the country is found, we attempt to locate the city the domain belongs to. The `whois.internic.net`, and `whois.ripe.net` usually provide the city where each domain belongs to, for US and European domains respectively. If the city is not found, the capital of the country is assumed to be the origin of the request. Once the originating city is decided, a local database in consulted, the latitude and longitude are found, and the request is displayed on the screen.

4. Related Work

Although there exists a significant amount of work in visualization (esp. for performance analysis of parallel applications) [2, 3, 4], visualization of world wide web traffic is a rather new topic.

Lam, Reed, and Scullin designed and implemented a real-time geographic visualization tool of world wide web traffic on top of the Pablo performance analysis toolkit, and the Avatar virtual reality software [1]. Although our work and [1] are very much related, we view them as complementary to each other. We see the focus of [1] to be on exploiting an existing toolset (Pablo and Avatar) into a new domain: WWW traffic visualization. On the other hand, our approach is on designing a simple, portable tool that can be easily used by web masters and users to understand the client base of their servers. Our tool is written in Java, and can be downloaded and used without any further requirements. In an environment where Pablo and Avatar are already up-and-running, it would be wise to use the tool described in [1]. On the other hand, in domains that have not installed Pablo and Avatar, our tool provides an easier way to visualizing WWW traffic.

Pitkow and Bharat implemented a tool, called WebViz [8], that visualizes WWW access logs. WebViz focuses on providing a graphical view of a web server’s local database and access patterns with the intention of answering the question: "How are people using the database?". Specifically, it displays the documents of the database and connections (links) between the documents as a web-like graph structure. Nodes in the graph represent documents, while the edges represent the hyperlinks between the document. A collection of edges is referred to as a path, that a user may has followed, while accessing the database. In addition to the graphical view, WebViz collects and provides information about the recency and frequency of access of each path and document. In contrast with WebViz, Palantir presents a

geographical visualization of the origin of the access requests. We believe that WebViz and Palantir are complementary to each other and may help WWW database designers and maintainers to take important decisions about the location of their web server and the structure of their database.

The idea of using a web-like graph structure to represent HTML documents and the hyperlinks among them has been used also by several other research groups with the aim of facing the "being lost in hyperspace" problem. The Navigational View Builder [9, 10, 11] is a tool that creates 2D diagrams representing the World Wide Web using various strategies, which reduce the problems of navigational graph development (understanding the context of a node from the diagram, and graph complexity). An algorithm that provides a way to give context in the nodes of a navigational diagram is presented in [10], while a way to reduce the graph complexity using multiple hierarchical views is presented in [11]. Muzner and Burchard provide another solution to the same problem by constructing graphical representations of the structure of sections of the World Wide Web in 3D hyperbolic space [12]. The representation has an hierarchical tree structure.

Recently, several other tools that visualize WWW information have been developed, which focus on displaying WWW information so that related documents are placed nearby in the displayed image. For example, the document exploration tool WEBSOM provides an ordered map of the information space is provided: similar documents lie near each other on the map. The order helps in finding related documents once any interesting document is found (`http://websom.hut.fi/websom/`). As another example, the *hyperspace* system allows a user to create a realtime visualization of the structure of a set of web pages while browsing through them. Its goal is to help the inexperienced users navigate the Web with ease [7].

Finally, Abrams, Williams, Abdulla, Patel, Ribler, and Fox have used CHITRA95, a tool able to visualize, and investigate collections of trace data from computer and communication networks, in order to explore the inter-access time of files in a server, the performance of a proxy server cache as well as the size and types of files requested [13].

5. Summary

In this paper we presented Palantir : a visualization tool that animates the source and amount of Web traffic in real time. The tool displays the origin and magnitude of a Web server's hits either in real-time or in batch mode. It can synthesize the traffic to several Web servers so as to get a global view of the hits in a multi-site organization. Palantir allows a user to ‘‘zoom” in and out the traffic at will. Using out tool, a user can get a deep understanding of *where* a server's clients are located, and thus how to reach them more effectively.

Palantir is written in Java and can be accessed on-line from `http://naxos.csd.uch.gr:9000`.

6. Acknowledgments

This work was supported in part by PENED project ‘‘Exploitation of idle memory in a workstation cluster” (2041 2270/1-2-95), funded by the General Secretariat for Research and Technology. We deeply appreciate this financial support.

The Computer Science Department of the University of Rochester provided us with some of the web server traces we display in this paper.

References

[1] S. Lamm, D.A. Reed, and W. H. Scullin. Real-Time Geographic Visualization of World Wide Web Traffic. *Computer Networks and ISDN Systems*, 28(7-11):1457-1468, 1996.

[2] Thomas J. LeBlanc, John M. Mellor-Crummey, and Robert J. Fowler. Analyzing Parallel Program Executions Using Multiple Views. *Journal of Parallel and Distributed Computing*, 9:203-217, June 1990.

[3] A. Malony, D. Hammerslag, and D. Jablonowski. Traceview: A Trace Visualization Tool. *IEEE Software*, 8(5):19-28, September 1991.

[4] B. P. Miller, M. Clark, J. Hollingsworth, S. Kierstead, S.-S. Lim, and T. Torzewski. IPS-2: The Second Generation of a Parallel Program Measurement System. *IEEE Transactions on Parallel and Distributed Systems*, 1(2):206-217, April 1990.

[5] JRR Tolkien. *The Lord of the Rings*. Harper Collins Publishers, 1989.

[6] A. Vahdat, P. Eastham, C. Yoshikawa, E. Belani, T. Anderson, D. Culler, and Michael Dahlin. WebOS: Operating System Services For Wide Area Applications. Technical Report UCB CSD-97-938, U.C. Berkeley Computer Science Division, March 1997.

[7] A.M. Wood, N.S. Drew, R. Beale, and R.J. Hendley. HyperSpace: Web Browsing with Visualization. In *Third International World-Wide Web Conference*, 1995. Available from http://www.cs.bham.ac.uk/~amw/hyperspace/www95/.

[8] J.E. Pitkow, and K.A. Bharat. WebViz: A Tool for WWW Access Log Analysis. In *First International World-Wide Web Conference*, 1994. Available from http://www1.cern.ch/WWW94/PrelimProcs.html.

[9] S. Mukherjea, and J. D. Foley. Visualizing the World-Wide Web with the Navigational View Builder. In *Third International World-Wide Web Conference*, 1995. Available from http://www.igd.fhg.de/www/www95/proceedings/papers/44/mukh/mukh.html.

[10] S. Mukherjea, and J. D. Foley. Showing the Context of Nodes in the World-Wide Web. In *Proc. of ACM SIGCHI'95*. Available from http://www.acm.org/sigs/sigchi/chi95/Electronic/documnts/papers/sm2bdy.htm.

[11] S. Mukherjea, J. D. Foley, and S. Hudson. Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views. In *Proc. of ACM SIGCHI'95*. Available from http://www.acm.org/sigs/sigchi/chi95/Electronic/documnts/papers/sm_bdy.htm.

[12] T. Munzner, and P. Burchard. Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. In *Proc. of VRML'95* (San Diego, California, December 14-15, 1995), special issue of Computer Graphics, ACM SIGGRAPH, New York, 1995, pp. 33-38. Available from http://graphics.stanford.edu/papers/webviz/.

[13] **13** M. Abrams, S. Williams, G. Abdulla, S. Patel, R. Ribler, and E.A. Fox. Multimedia Traffic Analysis Using CHITRA95. In *Proc. of ACM Multimedia 95* (San Francisco, California, November 5-9, 1995). Available from http://ei.cs.vt.edu/~succeed/95multimediaAWAFPR/95multimediaAWAFPR.html.