Visualization of Large Web Access Data Sets

Ming C. Hao, Pankaj Garg
Hewlett Packard Laboratories
Palo Alto, CA.
ming_hao@ hp.com
pankaj_garg@hp.com

Umeshwar Dayal, Vijay Machiraju
Hewlett Packard Laboratories
Palo Alto, CA.
Umeshwar_dayal@hplcom
vijay_machiraju@hpl.com

Daniel Cotting
ETH Swiss Federal Institute of Technology Zurich
Zurich, Swiss
dcotting@bluewin.com

Abstract

Many real-world e-service applications require analyzing large volumes of transaction data to extract web access information. This paper describes Web Access Visualization (WAV) a system that visually associates the affinities and relationships of clients and URLs for large volumes of web transaction data. To date, many practical research projects have shown the usefulness of a physics-based mass-spring technique to layout data items with close relationships onto a graph. The WAV system: (1) maps transaction data items (clients, URLs) and their relationships to vertices, edges, and positions on a 3D spherical surface; (2) encapsulates a physics-based engine in a visual data analysis platform; and (3) employs various content sensitive visual techniques - linked multiple views, layered drill-down, and fade in/out - for interactive data analysis. We have applied this system to a web application to analyze web access patterns and trends. The web service quality has been greatly benefited from using the information provided by WAV.

Keywords: Visual Clustering, Data Access, Similarity, Clients, Web Transactions

1 Introduction

Recently, the rapid increase of transactions on the Internet has led to the availability of large volumes of web transaction data. Business research efforts [1, 2, 3, 9, 11, 12, 13, 14] have focused on how to turn raw data into valuable information. For example, by exploring web data access behavior, business analysts are able to find and retain their most valuable users and evolve their best service strategies.

A web transaction starts with a user clicking on a web page. The client (web browser) sends the request through several components, such as applications servers, to perform some service. For example, a user clicks on a web page to purchase an airline ticket. The data access patterns through various components play an important role for the overall transaction. Often, it impacts the quality of end-user experience. In order to provide faster service, web analysts need to analyze the data and to balance the workload among their servers.

A common method for analyzing data access performance is to use scatter plots [10]. For visualizing web transaction data with a large number of clients, we have found that the scatter plot has too many overlaps. Only a small number (100-200) of low-density clients can be shown simultaneously. With large volumes of clients, the scatter plot quickly becomes cluttered and difficult to visualize, as illustrated in Figure 1A. (x-axis is number of the web clients; y-axis is the response time)

The following are the recent requirements for a new visualization system of large web transactions:

(1) Scale to a large number of web transactions and clients.
(2) Place clients with similar behavior and relationships close together.
(3) Unclutter the display with no overlapping.
(4) Interact with the user for analyzing different scenarios.

This paper describes a Web Access Visualization (WAV) system for addressing the above requirements in visualizing large volumes of web transactions. Section 2 describes the overall architecture of WAV. Section 3 describes our approach to visualization, which is a combination of (a) data mapping; (b) using a mass-spring technique; and (c) clustering. Section 4 describes how web analysts can perform various interactive analyses on WAV for different web applications. Section 5 summaries our conclusions.

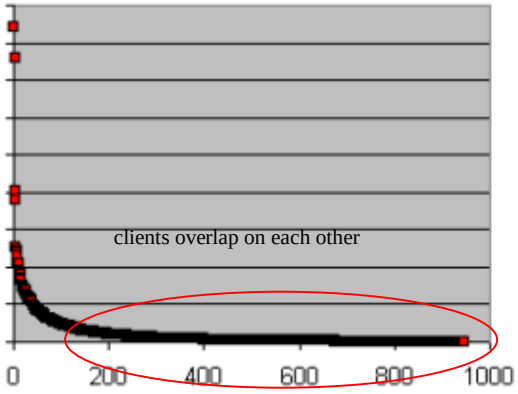


Figure 1A: A Scatter Plot (x-axis is number of the web clients; y-axis is the response time)

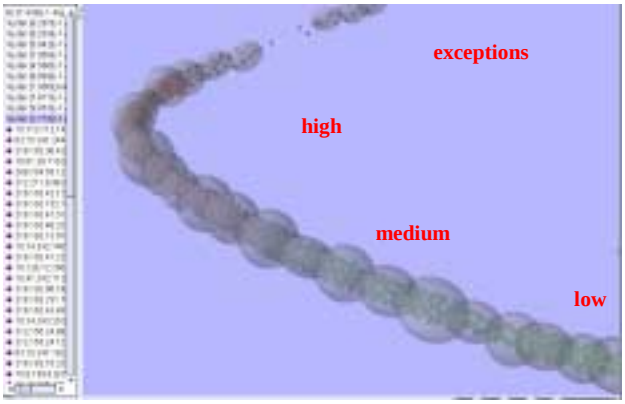


Figure 1B: A New Web Access Visualization
(rectangle represents client; color represents response time)

2 Using Similarity For Web Access Visualization

To analyze a large highly related web client space, we are experimenting with a new Web Access Visualization technique, called WAV. WAV uses similarity to place clients with similar data access patterns close together, as illustrated in Figure 1B. For example, WAV clusters the clients with a similar response time. The "distance" between each pair of clients represents the data access relationship. The most tightly related client is the client with the highest correlation with other clients. These clients usually have similar response times and access the same web pages.

The detailed methods and algorithms of WAV are described in the Hewlett Packard Technical Report [8], in which we describe the integration of a physics-based mass-spring engine and a data mining visualization system [7]. Also, we will describe how to cluster related web client transactions for pattern discovery. As to the scalability issue, we are experimenting with various methods to hide the complexity of the data.

The WAV system contains four basic components:

(1) Distance: the "distance" between each pair of items represents *the similarity of web access pattern.*
(2) Color: the color of the node is used to represent *the degree of the similarity, such as average response time*.
(3) Cluster: an ellipsoidal surface is used to wrap around highly related clients.
(4) Content sensitive visual techniques: linked multiple views, layered drill-down, and fade in/out methods are provided for interactive data analysis.

3 An Experimental Visualization System

Many practical research projects have shown the usefulness of a physics-based technique to layout data items with close relationships onto a graph. For example, the Ivory system [4, 5] has been applied to banking for analyzing the relationships among economic data. The DAV system [8] has been applied to market basket analysis for product recommendation. The current experiment on WAV is focused on the visualization of web access relationships.

The WAV system is built on Java-based multi-threaded parallelism. The WAV processor interacts with the user and extracts data from e-service engines or data warehouses for visually analyzing the data. The WAV processor manages the following three different processing states (As illustrated in Figure 2, 3, 4).

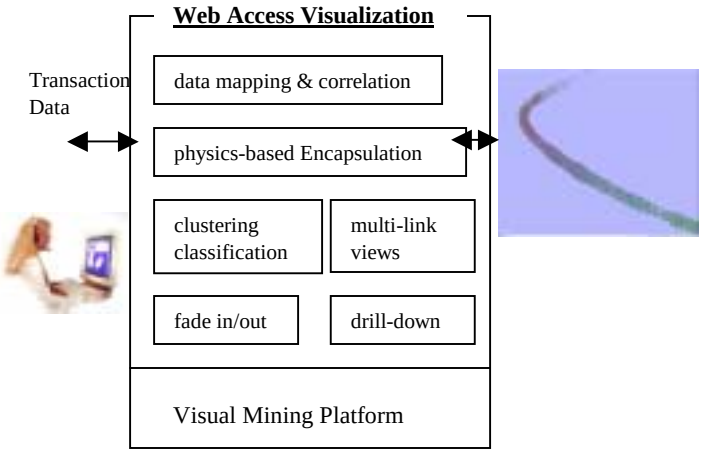


Figure 2: An Architectural Overview

The architectural components are described in sections 3 and 4.

3.1 Data Mapping and Correlation

WAV arranges the items (clients) extracted from the web transaction data onto a spherical surface. Items are represented as vertices. A transaction is a web log record that contains a client, a URL and network access response time. The transaction data is described as follows:

Transactions $\{T_1, T_2, \ldots, T_n\}$	1 to n = # of transactions
Items$\{I_1, \ldots, Im\}$	1 to m = # of data items (e. g., # of clients)
Item $I_i \rightarrow \{T_{i1}, \ldots, T_{ij}\}$	i=[1…m], j=[1…k] k<=n

WAV calculates the correlation between pairs of data items, based on some aggregate transaction metric, e.g., median response time (M). The data items will be laid out according to their similarity for the chosen transaction metric. The relaxation algorithm calculates the final layout. To create a more linear layout, the relaxation algorithm must be configured to perform the minimization process as accurately as possible.

For example, the similarity based on the difference in median transaction response times is calculated as below. WAV only stores similarity values with a difference that is less than δ (e.g., 20 percent) of the larger median access time (else 0 is returned). The reciprocal value of the difference is scaled by half the average difference value. The statistics can be arranged differently as needed. The computed results from the data similarity rule are stored in a data access matrix.

$M(I_i)$ = median response time for transaction involving item I_i

Similarity (I_i, I_j) =

$\min (1.0, (|M(I_i) - M(I_j)|/ \max (M(I_i)))$

3.2 A Physics-Based Mass-Spring Engine Encapsulation

The initial positions of items on the spherical surface could be at random. To avoid random pre-clustering, WAV distributes items equally on a sphere. The computation of equally spaced positions is based on a Poisson Disc Sampling [13] for approximation.
WAV encapsulates a physics-based mass-spring engine to connect web clients with springs. The strength of client relationships correlates to the stiffness of the springs between them. The stiffness of the spring is defined in a web access matrix. The mass-spring engine transfers the spring stiffness to the distance between pairs of clients. From the principal of a physics-based mass-spring engine, after many computations, the graph will be relaxed and reach a local minimum. Clients with close relationships are automatically moved together and form clusters.

Figure 3 illustrates the final graph after the graph has been relaxed using 500 iterations.

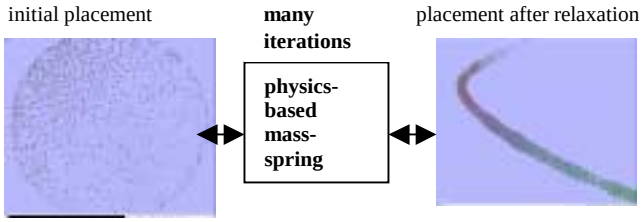


Figure 3: Encapsulate A Physics-Based Mass-Spring Engine

3.3 Clustering and Classification

After the energy of the graph layout is minimized as the result of the spring relaxation, WAV employs either K-mean or C-mean to group clients into different categories, such as fast, medium, or slow. WAV produces a clustered graph layout, as illustrated in Figure 4.

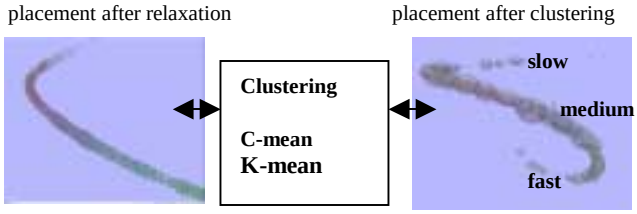


Figure 4: Cluster Highly Correlated Web Clients

4 Interactive Analysis and Applications

One common problem web analysts want to solve is to use web transaction history to improve client performance for faster service. They want to discover clients who causing network bottlenecks. The WAV technique has been used experimentally to visually analyze web client behavior with respect to response time for web pages at Hewlett Packard Laboratories.

Interactivity is an important aspect of a web access visualization system. To make large volumes of web transactions easy to explore and interpret, WAV provides the following interaction capabilities: (1) layered drill-down; (2) fade in/out; (3) multiple linked-views; and (4) an automatic alarm system.

Figure 5 (see color section) illustrates a series of graphs generated from a web transaction observation data set. It contains 35,000 transaction records. There are 986 clients with over two thousands URLs. The rectangles represent clients that make transactions on the web. The spheres represent URLs. WAV places clients with similar response times near to each other.

WAV employs the following interactive visualization techniques to allow web analysts to navigate the graph and to discover patterns and trends.

4.1 Layered Drill-Down

A drill-down technique allows the viewing of all related information after selecting a single data item. For identifying correlations, a subset of data items corresponding to related attributes can be highlighted in the WAV graph. After a WAV graph is generated, the web analyst can easily navigate the web client and URL distribution graph and answer questions, such as which are the fastest clients, which are the slowest clients, what types of web pages are being accessed the most and by which client?

Figure 5A (see color section) illustrates 986 clients arranged according to their response time, with the accessed over two thousand URLs placed around them. The web analysts can click on a client and find the answers to their questions.

4.2 Fade In/Fade Out

Figure 5B (see color section) shows the graph after fading out all the URLs. It allows the web analyst to focus on clients instead of URLs. The fade in/out mechanism allows users to categorize data items i.e. clients. For example, the web analyst is able to quickly locate the group of clients with response times in the range of 3,000 ms to 4,000 ms.

4.3 Multiple Linked Views

In many cases, the data to be analyzed consists of multiple relationships. With multiple linked views we can visualize correlations among data items. When multiple views are presented, items across all the views are linked. In Figure 5C (see color section), the data item in the graph is linked to the tree table at the left side of the graph. The web analyst can easy find both the average response for the cluster and each client's medium response time.

Figure 5D (see color section) illustrates six linked views to show the relationships among clients. The user may select a single cluster to find all the data items and their response times contained in the cluster. The six views are described as follows:

(1) A largest cluster with medium speed clients (highlighted as red)
(2) A cluster of slowest clients
(3) A cluster of fastest clients
(4) The categories of clients with respect to web response time
(5) The order of clusters from slow to fast
(6) Zoom in a cluster (highlighted as red) – drill down each member's detail information

4.4 Automatic Alarm System

Figure 5E (see color section) illustrates the WAV automatic alarm system. A WAV event will be triggered when an outlier client with either too small or too large access times is found. These

exceptional clients will be automatically highlighted and notified to the web analyst.

5 Conclusions and Future Work

Information visualization of web applications is an emerging technology that needs new techniques to visualize large volumes of massive transaction data with no overlaps. At Hewlett-Packard Laboratories, we have integrated a mass-spring system into a visual analyzing platform. We have used the system to visually analyze over a dataset containing 35,000 transactions with thousands of clients and URLs for web service analysis. WAV provides a fast and interactive way for web analysts to easily navigate through large volumes of web transactions to locate problems and to enhance web services. As a result, WAV has made a significant impact on web service quality. The quality of end-user experience has been greatly enhanced with fast services. Further research on issues such as scalability is continuing.

Acknowledgements:

Thanks to Sharon Beach, Aad van Morrsel, and Klaus Wurster from Hewlett Packard Software Technology Research Laboratory for their encouragement and thanks to Prof. Markus Gross from Swiss Federal Institute of Technology, Zurich, Switzerland for his suggestions and comments.

References

1. Stephen G. Eick, Joseph L. Steffen, Eric E. Sumner, Jr. “SeeSoft - A Tool for Visualizing Line Oriented Software Statistics”, IEEE Transactions on Software Engineering, 1992.
2. Thomas Gschwind, Kave Eshgi, Pankaj K. Garg, Klaus Wuster, “ Web Transaction Monitoring”, HPL-2001-62, March, 2001
3. A. Sahai, V. Machiraju, J. Ouyang, K. Wurster, “ Message Tracking in SOAP-based Web Services”, HPL-2001-199, August, 2001.
4. T. C. Sprenger, R. Brunella, M. H. Gross, “H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces”, IEEE/VIS2000.
5. T. C. Sprenger, M. H. Gross, “Ivory – An Object Oriented Framework for Physics-Based Information Visualization in Java”, IEEE InfoVis98, North Carolina.
6. M. H. Gross, T. C. Sprenger, J. Finger: “Visualizing Information on a Sphere”, Arizona, Proceedings of the IEEE Information Visualization 1997.
7. Ming Hao, Umesh Dayal, Meichun Hsu, etc. “A Java-based Visual Mining Infrastructure and Applications”, Proceedings of the IEEE Information Visualization 1999, CA.
8. Ming Hao, Umesh Dayal, Meichun Hsu, Thomas Sprenger, Markus H. Gross. “Visualization of Directed Associations in E-commerce Transaction Data”, VisSym01, 2000, Switzerland.
9. Giuseppe Di Battista, Peter Eades, “Graph Drawing Algorithms for the Visualization of Graph”, Prentice Hall, 1999.
10. Scatter Plot: Microsoft Excel
11. G. G. Robertson, J. D. Mackinlay, S. K. Card, “Cone Tree: Animated 3D Visualizations of Hierarchical Information, SIGCHI’91
12. John Lamping and Ramana Rao, “Laying out and Visualizing Large Trees Using a Hyperbolic Space”, ACM /UIST’94, 1994.
13. A. S. Glassner, “Principles of Digital Image Synthesis”, Morgan Kaufman, San Francisco, 1995.
14. Matthias Kreuseler, Norma Lopez, Heidrun Schumann, “A Scalable Framework for Information Visualization”, InforVis 2000, Utah.

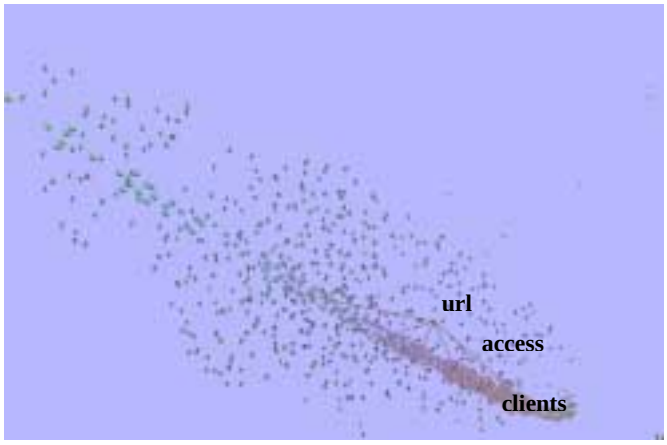


Figure 5A An Overall Layout of Web Clients and URLs

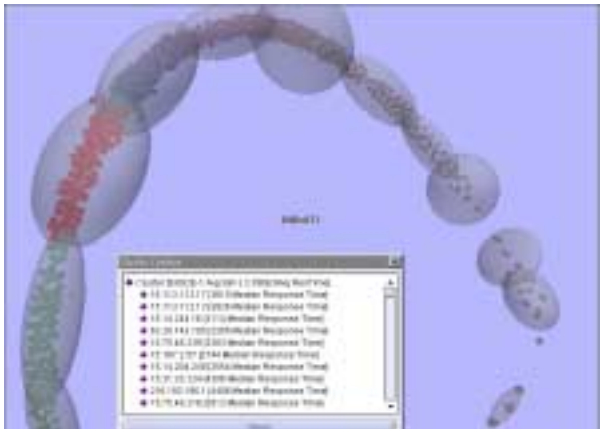


Figure 5B: A Client-Only Graph with URLs Fade -Out

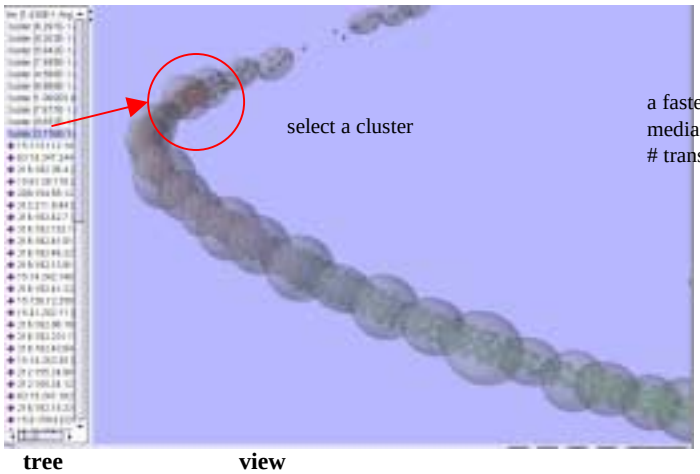


Figure 5C: Link a Selected Cluster To A Tree Table

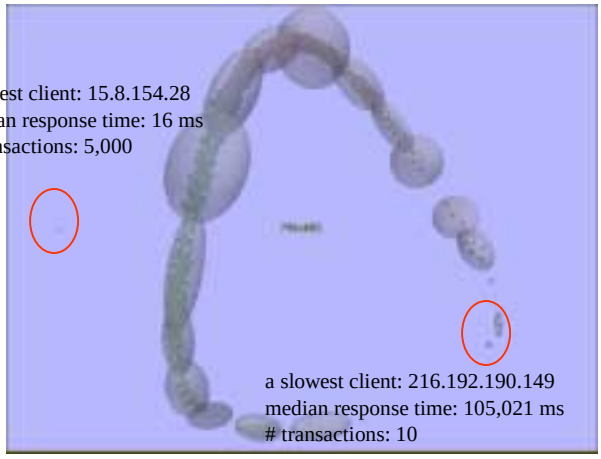
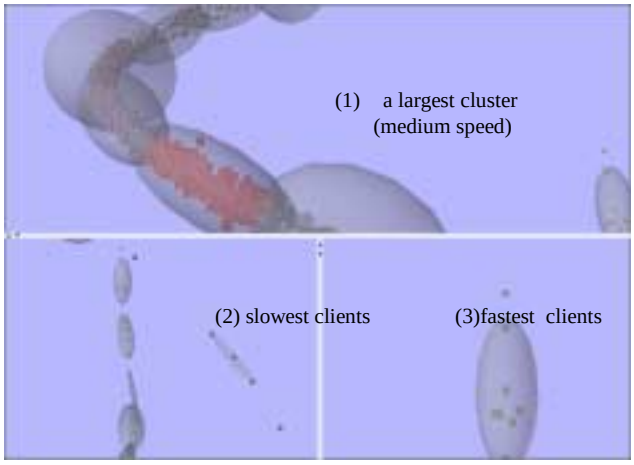


Figure 5E: Automatic Alarm System



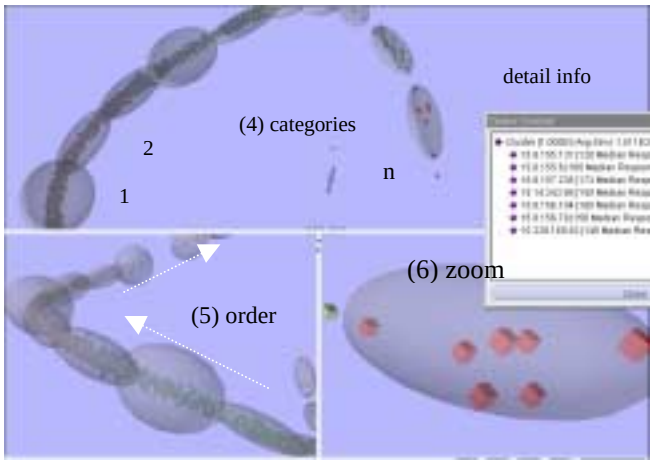


Figure 5D: Six Multiple Linked Views

Figure 5: An Example of Web Transaction Observation
(Analyze 35,000 web transactions with 986 clients, and over two thousands URLs)