VISUALIZATION OF GLOBAL WEB LOG AND WEB COMMUNITIES

Bowo Prasetyo, Shingo Otsuka, Masashi Toyoda, Masaru Kitsuregawa [1)]

Abstract

In the last decade Internet has emerged to be the largest information source in the world. Global traffic of millions people accessing Internet everyday is immense and may provide much interesting information. However, it is difficult to visualize and analyze global web access effectively due to its huge size. One obvious advantage of analyzing global web log over local server log is its ability to reveal the position of a website in the global Internet. Here, we have developed a tool to visualize Internet traffic data gathered from wide area of Internet users in Japan, and associate it to groups of related web pages, so called web communities. Using our visualization tool, we can effectively visualize global access patterns of multiple groups of websites around the target URLs to find where they are placed in the global web, as well as local access patterns inside a single group of websites to discover visitor behaviour within the sites. Additionally by the virtue of web communities data, we can reveal also user movement among web communities around the target URLs.

1. Introduction

With the explosive growth of information sources available on the World Wide Web, it has become increasingly necessary for users to utilize analysis tools in order to find, extract, filter, and evaluate the desired information and resources. In addition, with the transformation of the Web into the primary tool for electronic commerce, it is essential for organizations and companies, who have invested millions in Internet and Intranet technologies, to track and analyze their global presence in the Internet, as well as their local visitor access patterns. These factors give rise to the necessity of creating visualization systems that can effectively visualize information both across the global Internet and in particular Web localities.

Global web log contains user access data across multiple websites covering large area of Internet. This kind of log combined with web communities data make it possible for website administrator or ordinary Internet user to find:

1. Global position of particular websites among others/communities in the Internet.
2. Local visitor behaviour within particular websites.

1) Institute of Industrial Science, University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan

3. Relation and comparison to the rival websites.
4. Important websites/communities related to some keywords and how they are connected each other.

However, there are many difficulties to visualize user access patterns in global web log, including:

1. The log size is usually very large make it difficult to select relevant data in a manageable size.
2. Resulting graph often consists of hundreds even thousand of nodes which is impossible to display at once in a readable layout. Need more treatment.
3. The nodes are often connecting each other in a complicated manner yielding large number of edges, making the graph even more cluttered.

Here, we have developed a visualization system that can effectively visualize both globally and locally, information in the access log files collected from wide area of Internet users. Optionally we can also associate URLs to appropriate web community whose data is already prepared in advance. Using our tool we can visualize user access patterns globally across multiple groups of websites in the Internet to find where they are placed in the global web, as well as locally inside some particular group of websites to discover visitor behaviour within the sites.

The rest of this paper is organized as follows: Section 2 overviews some related works with its known problems. Section 3 presents Panel Log that is used in our experiment. Section 4 briefly describes the Web Communities data used. Section 5 and 6 explains several data filtering and visualization principles that we adopted in our tool. Finally, section 7 and 8 give experiment results and conclusions.

2. Related Works and Known Problems

In 2001 Hochheiser et. al. introduce a series of interactive visualizations that can be used to explore server data across various dimensions [4]. Interactive visualizations can be used to provide users with greater abilities to interpret and explore web log data. By combining two-dimensional displays of thousands of individual access requests, color and size coding for additional attributes, and facilities for zooming and filtering, these visualizations provide capabilities for examining data that exceed those of traditional web log analysis tools. While this visualization has advantages for analyzing statistical data, it cannot find users' access patterns among websites.

Prasetyo et. al. in 2002 introduced a tool for visualizing sequential pattern results from log files called Naviz [2]. This is a system of interactive access log files visualization that combines two-dimensional graph of visitor access traversals that considers appropriate web traversal properties, i.e. hierarchization regarding traversal traffic and grouping of related pages, and facilities for filtering

traversal paths by specifying visited pages and path attributes, such as number of hops, support and confidence. The tool also provides support for modern dynamic web pages. They applied the tool to visualize results of data mining study on web log data of Mobile Townpage, a directory service of phone numbers in Japan for i-Mode mobile internet users. Although their system has advantage to visualize web mining result patterns to discover interesting navigational behavior such as success paths, exit paths and lost paths inside a single website, it is not obvious if its technique can be applied too for visualizing users' access patterns among multiple websites.

3. Panel Log

Recently, similar to survey on TV audience rating, a new kind of business appeared in Japan, which collects URL histories of Internet users (called panel) who are chosen without statistic deviation. Global web log that we used is panel log files that are collected from thousands of panels through out Japan. While global web log is a collection of client side logs, it may has some similarity to a collection of web server access logs that are gathered from thousands of servers, in the sense that both of them contain global Internet user access data.

Log files are collected for period 2001-11-26 to 2002-09-01 (45 weeks) from about 10,000 panels. After sessionizing process (with 30 minutes timeout threshold) has been performed, the log contains 55,415,423 access records, 1,148,104 sessions and 7,777,955 distinct URLs (8.4 GB in size). The shortest user session contains 1 URL, the longest contains 6403 URLs, and average length is 48 URLs.

4. Web Communities

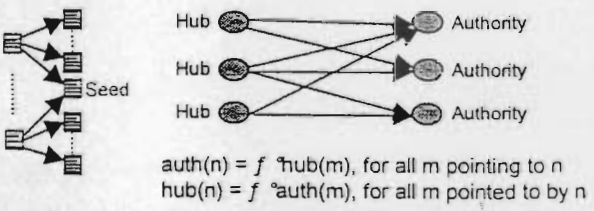

Figure 1. Related Page Algorithm

A web community is a collection of web pages created by individuals or any kind of associations that have a common interest on a specific topic, such as fan pages of a baseball team, and official pages of PC vendors. There are many techniques to extract web communities automatically from the web. The web communities data we used here is a work of Toyoda et. al. [5] that used hyperlink analysis to identify communities.

The algorithm mines web communities from a given seed set. The main idea is applying a related page algorithm (RPA) to each seed, then investigate how each seed derives other seeds as related pages. RPA first builds a subgraph of the Web around the seed, and extracts authorities and hubs in the graph using HITS as shown in *Figure 1*. Then authorities are returned as related pages.

To identify web communities and to deduce their relationships, the algorithm first puts focus on the relationship between a seed page and derived related pages. Consider that a page *s* derives a page *t* as a related page, and vice versa. This often means that both pages *s* and *t* are pointed to by similar sets of hubs. *If each page in a set derives other pages as related pages, the algorithm considers that this set of pages form a community.* Then, consider that a page *s* derives a page *t* as a related page, but not for the opposite. This means that *t* is pointed to by many different hubs such that *t* derives a different set of related pages excluding *s*. *In this case, the algorithm considers that community of page t is related to the community of page s.* Communities found are then labeled using several most frequent anchor texts used to link URLs inside.

5. Data Filtering Principles

Global web log used in our research contains 1,148,104 sessions that we would visualize using our tool. Well written visualization program using good graph layout algorithm will ideally be able to display virtually unlimited number of nodes. However, Internet user accesses are usually complicately connected nodes that will produce large number of edges, which in turn will dramatically reduce the number of nodes that are possible to display in a readable manner. As a consequence, we should employ some data selection/filtering methods such that we may choose only the most important data to display.

Our tool is utilizing many data filtering techniques, which may be categorized into two groups: database level filtering and application level filtering.

5.1. Database Level Filtering

Powerful and efficient database retrieval method is very essential in our visualization tool to provide good starting point for subsequent procedure. To retrieve data using keywords from database, we employ some techniques as described below:

1. Since session database is very large, it is not efficient to apply keyword matching in it directly. Instead, we will create URL database that consists of all URLs found in all sessions. This URL database will have size much less than session database and contain several important properties such as URL and its title to do keyword matching, session count that is aggregated from all sessions containing the URL, and session list where the URL is contained.
2. Retrieving all matched data is not a wise idea, since its number can be as large as tens of thousands of sessions that are impossible to visualize and difficult to handle. We should have some mechanism to pick only the most important data. Here, the session count in URL database does the task as importance measure. While doing keyword matching in URL database, we sort

the results by session count descending, such that most important data will come first, then we can cut the top results as needed.

3. In our log a session may contain as few as a single URL or as many as thousand URLs. Therefore, it is needed a filtering mechanism such that we can retrieve all sessions containing target URL without having to bring thousands of unnecessary other URLs in session into memory. Only several neighboring URLs "near" the target URL in the session should be retrieved and displayed, not the entire session. Here, we define "near" by specifying the distance (hop number) between target URL and neighboring URLs. If the neighboring URLs appear before target URL in session, then we call the distance "in-hop number", and for the opposite we call it "out-hop number". We call this technique *in-out filtering*. This way we can retrieve both short sessions as well as long sessions effectively.

Utilizing this searching techniques we are able to retrieve from database only the most important data that is relevant to user supplied keywords.

5.2. Application Level Filtering

Once the data is retrieved from database, it is stored in memory so we can apply further procedures before we begin visualizing it. These subsequent procedures include abstraction, focusing, layout generation, and interactive application level filtering. The first three procedures is purely visualization techniques and have nothing to do with data filtering, so they will be explained in the next visualization principles section.

Our application level filtering is needed to further squeeze the number of data should be displayed, as well as give flexibility to users in interacting with data such that they could manipulate the visualization in interactive manner. Application level filtering methods include:

1. Visitor Number Filtering: shows only nodes whose visitor number is larger than some specified threshold. Visitor number is aggregated on the fly while building layout using session count found in URL database. Here, we can specify different thresholds each for target nodes and its surroundings.
2. Edge Strength Filtering: shows only edges whose strength is larger than some specified threshold. We define edge strength by the number of sessions that pass through the edge. Edge strength is summed up on the fly when the graph is constructed from retrieved data.
3. In-Out Filtering: the same in-out filtering such as database filtering is used when displaying nodes on the screen. This way we can avoid graph cluttering due to long sessions.

These application level filtering methods greatly reduce the number of nodes and edges that should be displayed, as well as give users the freedom and flexibility to control the amount of information they

want to visualize and analyze.

6. Visualization Principles

Alberto O. Mendelzon in [1] discussed about several principles on visualizing World Wide Web. These are layout, abstraction, focus and interaction.

6.1. Layout

The simplest approach to Web visualization is that the Web is a graph. Currently there are many algorithms for drawing graphs such as GraphViz [3], spring embedder [6], simulated annealing [7], and TouchGraph [8]. We could pick one (or several) and use them to draw some portion of the Web. However, this alone does not work as expected.

Currently we are adopting graph-drawing algorithm based on TouchGraph [8], an open source graph-drawing algorithm that is a kind of spring-based algorithm. A node in the graph may represents URL, domain name or web community depending on user choice. A directed edge represents visitor traversal from tail to head node. Node color expresses its visitor number and edge color expresses its strength. Color ranges continuously from blue to red for low to high visitor number or edge strength.

One of the problem is scale. The Web consists of millions websites which are connected one to another by the means of hyperlinks physically or user traversals abstractly. The challenge is how we can visualize as much as data while keeping the layout as simple as possible, such that we could visualize and understand easily and comprehensively the information that we interested in. Therefore, our layout algorithm should be able to put large number of nodes and edges on the graph without cluttering.

Our TouchGraph based algorithm has several features that make it suitable for our purpose:

1. It automatically and evenly distributes nodes and edges in the radial direction if possible.
2. It draws graph on arbitrary width and height, so it can handle virtually unlimited number of nodes and edges, because graph size is no longer bounded by physical screen size.
3. It can scroll between the screen from one part of graph to another, and zoom in and out on any particular part of graph, so users can analyze graph in its full size as well as in detail.

6.2. Abstraction

Abstraction is an obvious technique for making complex networks manageable by imposing structure on what looks like chaos. Abstraction often goes under the name of clustering or grouping in the

literature. Nodes often can be grouped together into higher level "clusters" based on some appropriate criteria.

In our system basically we use two kinds of abstraction, i.e. grouping by domain name and web communities. When building graph layout, user can choose either to group URLs into its domain name or into web communities, to visualize global access pattern among multiple groups of websites. Alternatively user may also choose not to group URLs at all to visualize local access pattern within a single group of websites. Another additional abstraction is to treat URLs beginning with "www." same as its counterpart without the leading "www.". This technique can further simplify the resulting graph without reducing the quality.

6.3. Focus

One way of focusing is to select for display only information that is relevant to the task at hand. Since we deal with a large number of data, providing powerful ways to focus on relevant information is very essential in our tool. Our approach is to integrate with the visualizer some mechanisms to filter the information being visualized. As explained above, we utilize both database level filtering as well as application level filtering mechanisms to achieve the purpose.

Another way of focusing is:

1. Emphasizing certain parts of the display, while retaining the rest in de-emphasized form to provide context. For example, we applied this approach by making the color of target URLs to be more stand out than its surroundings, such that it can be recognized easily where the information of our interest is located in the graph.
2. Collapsing/expanding edges of a node.
3. Hiding a single node or a group of nodes. For example we may want to hide search engine nodes as they are usually massive and dominate almost the entire graph.
4. Selecting a group of nodes while hiding others etc.

6.4. Interaction

There are many ways in which direct manipulation can make Web visualization more informative. Here, we use many techniques to interactively control the visualization system. As mentioned in the previous data filtering principles, other than DB level filtering, we use also application level filtering which is applying interactive methods. Like wise other focusing techniques as described in the previous section are all using interactive methods. Another interaction technique includes "hovering" the mouse over a node that has the effect of showing the node info such as full URL, title, visitor number etc.

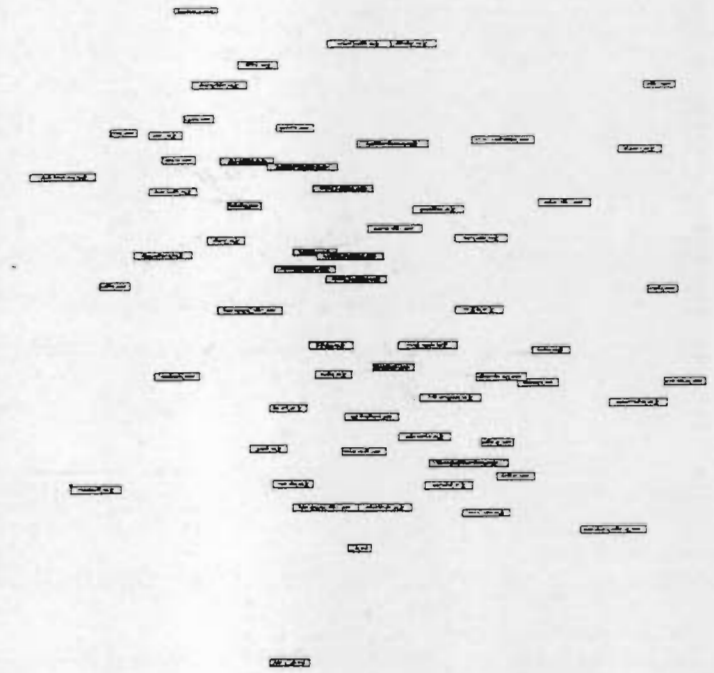

Figure 2. Global access patterns in full size.

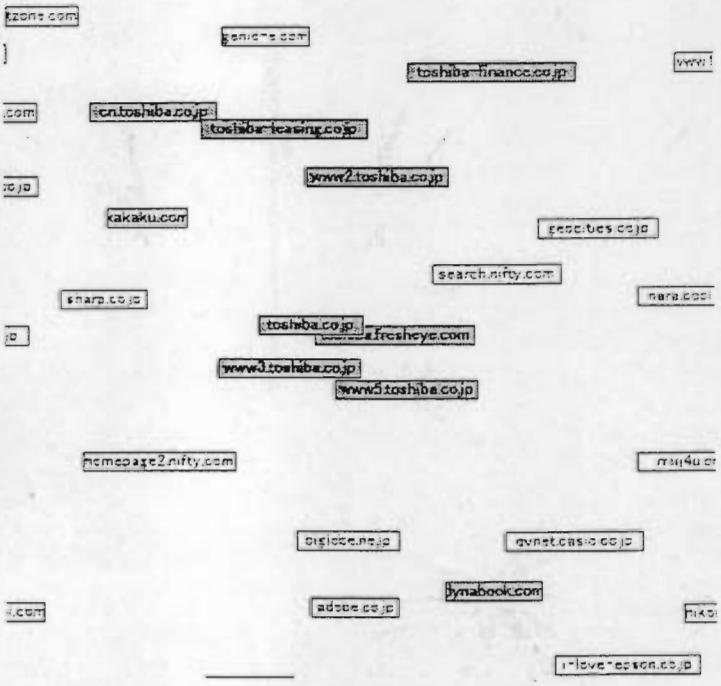

Figure 3. Global access patterns in zoomed out view.

7. Experiment Results

7.1. Global Access Patterns

For example we will use "toshiba" as keyword to search the database and show its global access patterns. First, we should limit the number of target URLs to be retrieved, we set that top 1000 URLs (regarding to visitor number) whose URL or window title containing word "toshiba" should be retrieved. In and out hop number is set to 10 hops for DB level and 1 hop for application level filtering.

This resulted in 1371 nodes and 3421 edges being constructed in memory. *Figure 2* is the full size version of results using domain-name abstraction and displaying 84 nodes and 139 edges using edge strength threshold of 3. *Figure 3* shows part of the zoomed out view of the results. In this result we can see that among top domains for keyword "toshiba" are 'toshiba.co.jp'[2] and 'dynabook.com'[3]. Further we may check that while 'dynabook.com' having many various connections to other domains, such as 'kakaku.com'[4] and 'i-love-epson.co.jp'[5], the main site 'toshiba.co.jp' only have one or two important outside connections. This may imply that 'dynabook.com' is more important and becoming the main destination for most people rather than 'toshiba.co.jp'. And site administrator could try to improve 'toshiba.co.jp' site so it will have the same popularity as 'dynabook.com'.

Applying web community abstraction we got 1450 nodes and 3720 edges in memory, and 72 nodes and 170 edges on screen using edge strength threshold of 4. *Figure 4* is part of the zoomed out view of the results. From this result we can see some communities around the sites, for example: "link special/affiliation/organization/collection" is mostly coming to 'toshiba.co.jp' [6], and "mistake/criticism/car/mycar/market price/used car/auction" to 'dynabook.com' etc. So we can know which communities are interested to our site which are not.

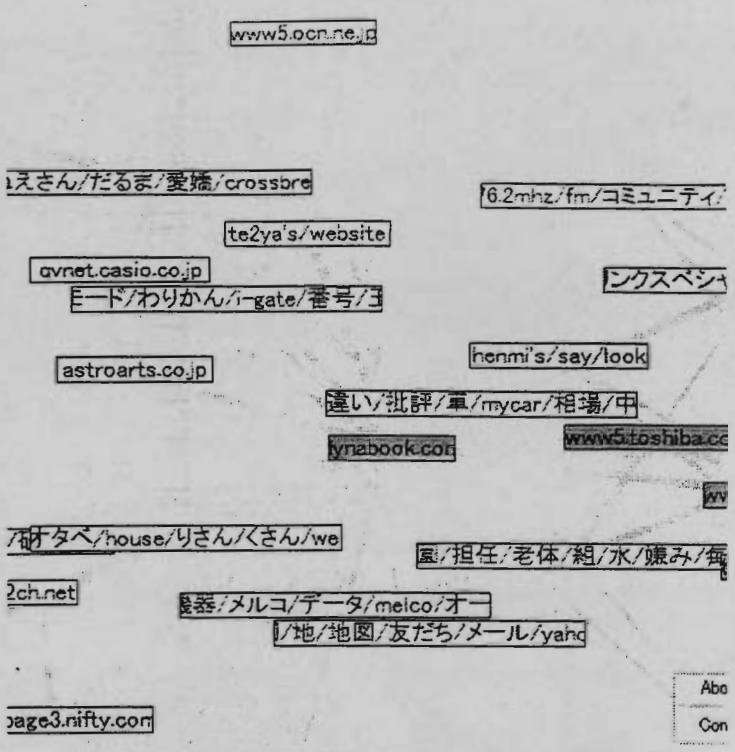

Figure 4. Global access pattern grouped by communities

7.2. Local Access Patterns

After we know how Internet users are moving around keyword 'toshiba', we may want to visualize also local access patterns inside the group of 'toshiba'. We could do this by choosing not to do URL abstraction at all when building graph layout such that all nodes will represent individual URLs. And to achieve a cleaner and larger graph, we will set in-hop and out-hop number to 0 such that only target URLs will be displayed as shown in *Figure 5*.

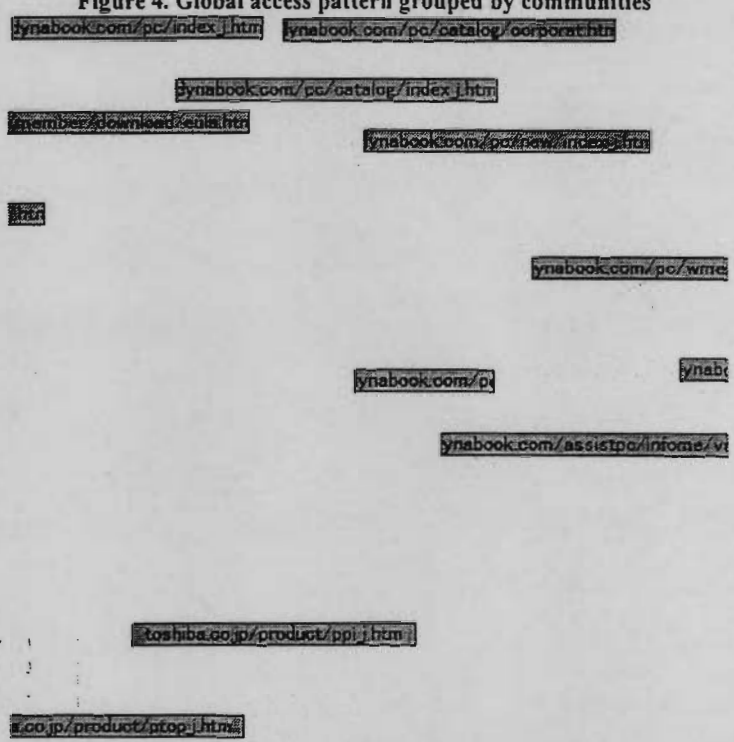

Figure 5. Local access pattern of group 'toshiba'

---

[2] Toshiba Group's official site contains inquiry, help, support, on-line shopping, press releases etc.
[3] Dynabook's official site contains Note PC, TV, DVD, peripherals etc.
[4] Site about shopping guidance, price comparison for computer, laptop, palm PDA, television etc.
[5] Official site of Epson contains printer, scanner, camera, product information, support, download etc.
[6] Translated from the original Japanese community label.

Checking the connection in and out of each nodes, we found that the main top pages include: 'dynabook.com/pc', 'toshiba.co.jp/index_j3.htm', and 'www5.toshiba.co.jp/pcss/member/download/index_j.htm'. From these top nodes, we can further track the paths to other nodes that we interested in.

8. Conclusions and Future Work

To be able to visualize global web log effectively, powerful data filtering methods combined with the right graph layout that can display a large number of nodes and edges is a necessity. This should be accompanied by appropriate abstraction methods to reduce the number of nodes, as well as focusing and interaction methods which provide users greatest flexibility to control visualization of data.

In the future we plan to expand tool's filtering functionality such as session filtering so that it can show only individual session selected by user. Local access patterns may be presented using different layout algorithm to provide more intuitive result. And we plan also to introduce user search words analysis whose data is available in the original log files.

9. References

[1] ALBERTO O. MENDELZON. Visualizing the World Wide Web. In Proceedings of the International Workshop AVI '96, pages 13--19, Gubbio, 1996.

[2] BOWO PRASETYO, IKO PRAMUDIONO, KATSUMI TAKAHASHI and MASARU KITSUREGAWA. Naviz : Website Navigational Behavior Visualizer. Pacific-Asia Conference on Knowledge Discovery and Data Mining, 276-289, 2002.

[3] E.R. GANSNER, E. KOUTSIFIOS, S.C. NORTH and K.P. VO. A Technique for Drawing Directed Graphs. IEEE Trans. Software Engg., 19:214--230, 1993.

[4] HARRY HOCHHEISER and BEN SHNEIDERMAN. Using interactive visualizations of {WWW} log data to characterize access patterns and inform site design. Journal of the American Society of Information Science, 52, 4, 331-343, 2001.

[5] M. TOYODA and M. KITSUREGAWA. Creating a Web Community Chart for Navigating Related Communities. In Conference Proceedings of Hypertext 2001.

[6] P. EADES. A heuristic for graph drawing. Congressus Numerantium, 42:149--160, 1984

[7] R. DAVIDSON and D. HAREL. Drawing graphics nicely using simulated annealing. ACM Trans. Graph., 15(4):301-331, 1996.

[8] http://touchgraph.sourceforge.net/