webSPADE: A Parallel Sequence Mining Algorithm to Analyze Web Log Data

Ayhan Demiriz
E-Business Department, Verizon Inc.,
919 Hidden Ridge, Irving, TX 75038
E-mail:ayhan.demiriz@verizon.com

Abstract

Enterprise-class web sites receive a large amount of traffic, from both registered and anonymous users. Data warehouses are built to store and help analyze the click streams within this traffic to provide companies with valuable insights into the behavior of their customers. This article proposes a parallel sequence mining algorithm, webSPADE, to analyze the click streams found in site web logs. In this process, raw web logs are first cleaned and inserted into a data warehouse. The click streams are then mined by webSPADE. An innovative web-based front-end is used to visualize and query the sequence mining results. The webSPADE algorithm is currently used by Verizon to analyze the daily traffic of the Verizon.com web site.

1 Introduction

This paper introduces an algorithm to analyze click streams from raw web logs. Click streams are collections of hits from specific user sessions. Assuming that user sessions in web logs are constructed by appropriate technology, we must first clean the web logs to remove redundant information. Parsing the cleaned web logs and inserting the data into a repository (data warehouse or relational database) is the next step in the analysis process. Data stored in a repository is easily used for frequency analysis with proven database technologies to create excellent summary reports. However, when it comes to analyzing the sequences, even with well defined process flows, the number of nested queries required to follow the processes step by step within a relational database framework makes the analysis prohibitively expensive. This expense, combined with the fact that simple database queries are unlikely to discover hidden sequences and relationships within the data, make it important to use an effective sequence mining algorithm to analyze the data contained within the web logs.

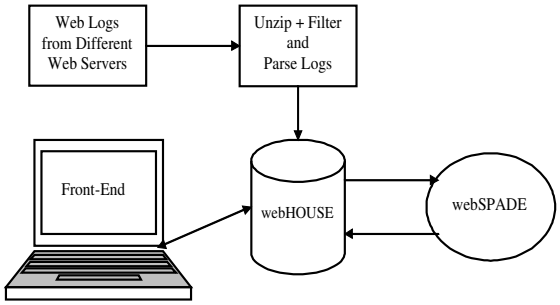
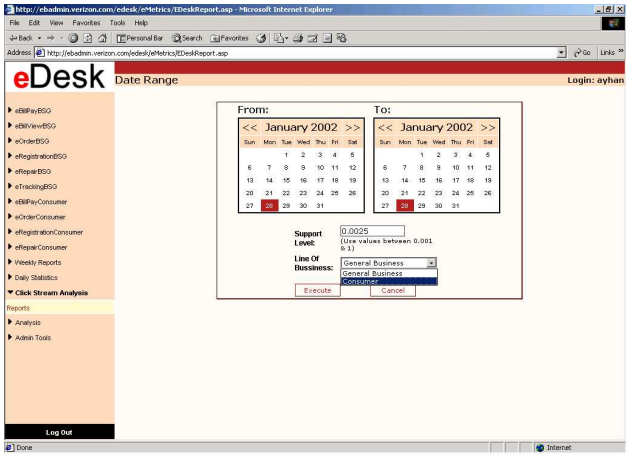


Figure 1. System Architecture

We propose a parallel sequence mining algorithm based on [3, 4]. There are several major differences in this paper compared to earlier work [3, 4]. Differences and improvements can be summarized as follows: 1- webSPADE is a Wintel-based parallel implementation. 2- It only requires one full scan of the data compared to three full scans in previous algorithms. 3- Temporal joins are used in webSPADE contrast to the non-temporal joins in the original algorithm. 4- The design of webSPADE achieves data and task parallelism simultaneously. 5- The current system has been in production since Mid-October of 2001 without any major problem. 6- Click stream data is analyzed daily and sequences are stored in a relational database. 7- A user-friendly front-end is used to visualize and mine stored sequences for a user-determined time range and support level. 8- By using front-end, it is possible to analyze click-



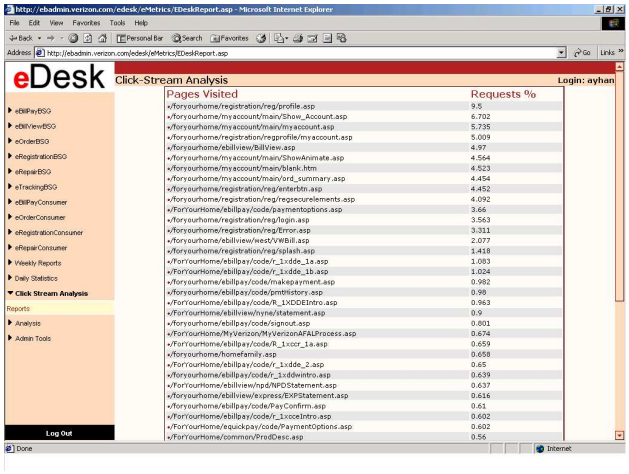
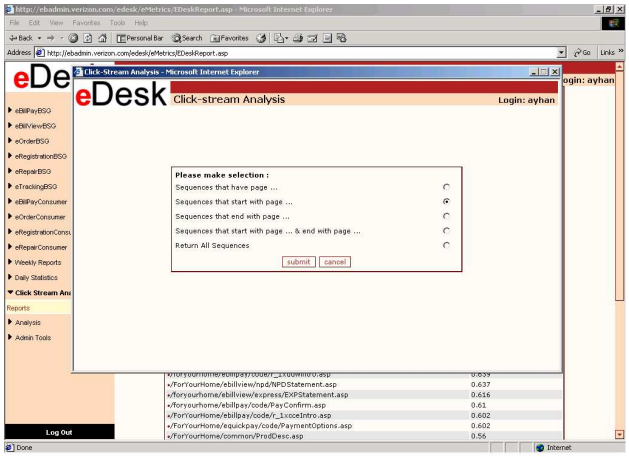


Figure 2. (A)- Parameter Selection Screen, (B)- Frequent Pages



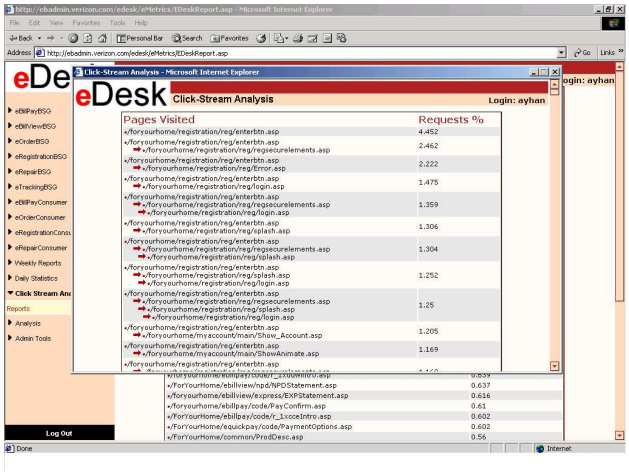


Figure 3. (A)- Analysis Selection, (B)- Sequences

stream data from a very large time period (e.g. whole year) in a short time.

We propose an integrated solution for performing click stream analysis in this paper. A simplified system architecture is shown in Figure 1. The parser feeds the data into webHOUSE, a data warehouse. The sequence mining algorithm, webSPADE, reads daily data from webHOUSE and inserts the daily sequences into webHOUSE. The front-end is used to query the webHOUSE to analyze sequences.

The nature of the sequence mining problem makes massive computation unavoidable. This situation is the ideal application for parallel programming. Parallel sequence mining algorithms are generally derived from sequential ones by introducing load balancing schemes for multiple processors and distributed memory. Since the data warehouse is built on MS SQL Server in our implementation, webSPADE has been developed in the Wintel environment, simplifying the parallelization of the serial programs.

The core point of our implementation is to modify the SPADE algorithm for the purpose of performing click stream analysis. The details of the original SPADE algorithm can be found in [3]. The main advantage of using this algorithm is the use of join operations instead of scanning all the data to count certain item sets. The original SPADE algorithm proposed in [3] requires three full scans of the data. Note that both SPADE and webSPADE may require large number of scans on intermediate partial data.

The paper is organized as follows: We introduce an analysis oriented front-end in Section 2. An analysis of web log data from Verizon.com during the month of January of 2002 is given in Section 3 for illustration purposes. Computational times are also reported in Section 3. The paper wraps up with a conclusion. Interested readers are advised to read the full version of this paper available at `http://www.rpi.edu/~demira/research.htm`.

2 Front-End

We present a very innovative way of scaling up any sequence mining algorithm by utilizing the database technology to analyze very large datasets spanning a large time window e.g. a quarter. A user friendly front-end enables users to choose three different parameters: support levels, time window(start and end date of analysis) and line of business (LOB) to query the sequences. Similar visual pattern analysis techniques are introduced in [1, 2] in the context of the "mine and explore" paradigm using episode and association mining.

Our approach is very simple. webSPADE runs **daily** with a predetermined support level to find sequences based on different lines of business; Support level is set to 0.1% for General Business and 0.25% for Consumer in our application. **Daily sequences** are then stored in a relational table. By design, webSPADE limits the length of sequences and, in this particular application, the maximum length of sequences is set to ten. Thus the relational table is composed of ten fields to determine the sequences, analysis date, line of business, frequency of sequence and total hits on analysis date.

Stored sequences can be used to analyze click streams between user specified dates. As seen from Figure 2(A), users can specify any support level and dates for a given line of business allowing a great degree of flexibility. **It should be noted that webSPADE is run daily and results are stored; there is no on-the-fly computation when the user selects parameters using the parameter selection screen**. In fact, stored results are aggregated and presented to the user. This point is the major difference from the previous work introduced in [1, 2]. On the other hand, our methodology requires mining efforts on a portion of data (for one day). Resulting patterns are then aggregated and presented to the user for visual analysis and pattern search. Moreover, our approach scales up the underlying mining algorithm and visualizes the results simultaneously. This is a significant improvement in terms of computation efficiency.

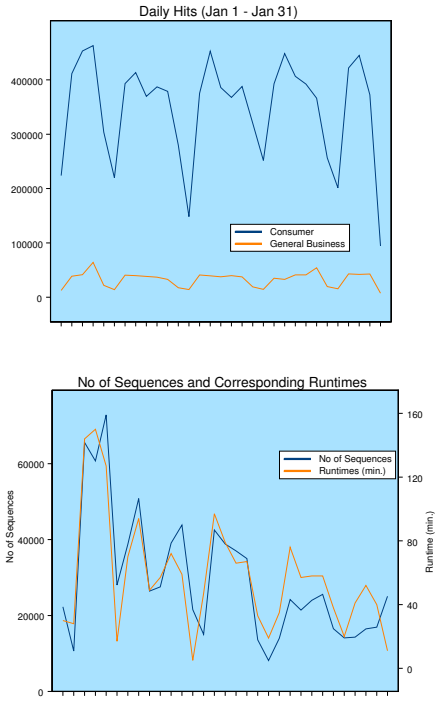


Figure 4. (A) Daily Hits, (B) Number of Sequences and Runtimes

After selecting the parameters, frequent pages are shown to the user as seen in Figure 2(B). Further analysis can be deployed by clicking on any page name in frequent page list (see Figure 2(B)). Once clicked, a pop-up window appears as seen in Figure 3(A). There are five options to choose on this screen. Users can select to see sequences that contain a selected page, start with a selected page, or end with a selected page. Users can list all the sequences as well. There is another option to list all the sequences starting with the selected page and ending with another page. Each option corresponds to a different stored procedure in the database. Having results stored in a relational table gives tremendous flexibility to report and query the results. Once an option is chosen, the resulting sequences are listed in the browser window as seen in Figure 3(B).

webSPADE and the web-based front-end, depicted in Figure 2 and 3, can be used for other time dependent sequential data. To illustrate the practical usage of sequence mining and to assess the performance of webSPADE we analyze all the data from January 2002 in next section.

3 An Illustrative Example

webSPADE has been used for analyzing web log data since mid-October of 2001. We can now analyze virtually all the frequent sequences since then by using the front-end reporting tool mentioned in the previous section. So far there are approximately 480M cleaned hits in the datawarehouse. Based on this data, webSPADE has found approximately 6M frequent sequences. To illustrate the usage of the webSPADE algorithm, we consider the data from January 2002. As we mentioned above, certain pages on Verizon.com are tagged to collect session information and our analysis covers only these pages. As we also mentioned above, the pages of two different lines of businesses are analyzed separately. Total daily hits during January are depicted in Figure 4(A). When we consider the whole month, there are approximately 12 million hits (requests) from both lines of business. This is a relatively large dataset.

The number of sequences and daily runtime of webSPADE in minutes is depicted in Figure 4(B). Reported times are the sum of runtimes for both lines of business. As expected, the number of sequences depends on number of daily hits (requests). Although webSPADE is not run on a dedicated server, the example runtimes of webSPADE can be considered close to reality. Note that runtimes also include both database access time and insertion time of sequences into the relational table.

The operational value of sequence mining is undeniable. For example, it is easy to monitor the traffic to understand whether a web page is functioning well or not. More specifically, certain pages might experience heavy traffic but the following pages may experience very low traffic. Sequence mining can easily catch such patterns. Some design problems might cause such patterns e.g. a misplaced next button at the bottom of the page; many people may not be able to see the next button because of smaller monitors. Such changes are requested in the light of sequences mining findings.

Sequence mining can also be used to find out who comes to a certain page and where they go after that page. For example, although bill view and bill pay processes are independent from each other in General Business pages, a significant portion of customers first view their bill and then pay it in the same session. It is also found that the bill view process sometimes fails to show bills due to the database access timeout. So, if we can increase the reliability of the bill view process, some of our customers will pay their bills in the same session. This is a very simple conclusion but it might take some time to come up with plain SQL analysis. Large scale sequence mining enables us to come to a conclusion on this matter more rapidly.

4 Conclusion

We successfully applied a parallel sequence mining algorithm to perform click stream analysis. webSPADE requires only one full scan of the database, but several partial scans of the database. Data and task parallelism are easily achieved using multi-threaded programming. Load balancing is left to the operating system. Post-implementation tunings are still ongoing to speed up the process even more. An innovative analysis technique to scale up sequence mining algorithm is also used for value-added business analysis.

Acknowledgments

I would like to thank Dr. M. J. Zaki for his helpful comments and directions on improving this paper and work.

References

[1] M. Klemettinen, H. Mannila, and H. Toivonen. Interactive exploration of discovered knowledge: A methodology for interaction, and usability studies. Technical Report C-1996-3, Department of Computer Science, University of Helsinki, 1996.

[2] M. Klemettinen, H. Mannila, and H. Toivonen. Interactive exploration of interesting patterns in the telecommunication network alarm sequence analyzer tasa. *Information and Software Technology*, 41:557–567, 1999.

[3] M. J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning Journal*, 42(1/2):31–60, Jan/Feb 2001. Special issue on Unsupervised Learning (D. Fisher, editor.).

[4] M. J. Zaki. Parallel sequence mining on shared-memory machines. *Journal of Parallel and Distributed Computing,*, 61(3):401–426, March 2001. Special issue on High Performance Data Mining (V. Kumar, S. Ranka and V. Singh, editors.).