Journal Of Harmonized Research (JOHR)

Journal Of Harmonized Research in Engineering
6(4), 2018, XX-XX

ISSN 2347 – 7393

Original Research Article

ANALYSIS END USER BEHAVIOR OVER WEB SCENARIO FOR EFFICIENT PREFETCHING THROUGH BIG DATA ANALYSIS

Mr. Anil Nayak, Mr. Shivendra Dubey, Mr. Umesh Joshi, Mr. Mukesh Dixit

Department of Computer Science & Engineering
Radharaman Engineering College, Bhopal (M.P.),India

Abstract: Web caching and restoration are the most common techniques that play a key role in improving web performance by keeping web objects that are likely to be visited in the near future closer to the customer. Web caching can work independently or in combination with Web pre-fetching. Cache and Web fetching can complement each other as the temporary Log file is a crucial part of web application. The log analysis is an important issue for the web application. Log file is not to be over emphasized as a source of information in systems and network management, whereas conduct efficient investigation and gathering of useful information need to correlate different log file. Task of analyzing event log files with the ever-increasing size and complexity of today's event logs has become cumbersome to carry out manually. Nowadays latest spotlight is automatic analysis of these logs files. Analysis of Web log is an innovative and unique domain constantly formed and modified by the convergence of several emerging web technologies. Because of its interdisciplinary nature, the diversity of issues addressed, the variety and the number of Web applications, is subject to many different methodologies. Proposed methodology has been used as effective web pre-fetching technique for frequent page generation by using relative position by inter session gap. The fast and efficient web pre-fetching scheme has improved user response of web page and expedites users visiting speed.

Keywords: Web mining, web caching, web prefetching, Clustering, Client server Architecture, Log File.

For Correspondence:
143anil.nayak@gmail.com
Received on: XXXX 2018
Accepted after revision: XXXX 2018
Downloaded from: www.johronline.com
DOI: 10.30876/JOHR.6.4.2018.XX-XX

Introduction: Rapid growth in Web application has increased the researcher's interest at this time. Everyone is surrounded by a computer network. A Web Application is a very useful application used to communicate and transfer data. An application that is accessed through a web browser is called the Web application network. Web caching is a known strategy to improve the performance of

a web-based system by saving Web objects that are likely to be used in the near future in a location closer to the user. Web cache mechanisms are implemented at three levels: client level, proxy level, and original server level [1,2]. Significantly, proxy servers perform essential functions between users and websites by reducing user response time and network bandwidth. Therefore, to achieve a better response time, an effective caching policy must be built on a proxy server. Web applications typically use a combination of server-side scripts (ASP, PHP, etc.) and client-side scripts (HTML, JavaScript, etc.) to develop the application. The client-side script processes the presentation of the information, while the server-side script handles all difficult things, such as storing and retrieving information [1,3].
The Internet is an essential resource for sharing information around the world. It has a lot of news, advertising, global communication between people and a lot of knowledge for students. This tremendous use of the Web or WWW makes it more important in the search world. The researcher faces the challenge of making web applications more efficient. Many researchers work on it and give a new idea to give the best results from the previous one. This message puts the best you have forward in this age. There is a great need to improve server response time for web applications. The current web contains a huge repository because it suddenly increases its use. It should focus on the quality and quantity of web content. Even when the speed of the Internet improves with lower costs, traffic gets heavier. Huge information makes it difficult to find relevant information quickly. This has led to efforts to improve speed, by reducing response time, and making the web more relevant.

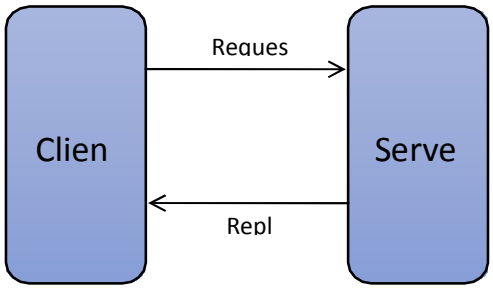

Figure 1: Client Server Architecture

Cache pre-fetching plays an important role in improving response time and making the application well organized. Pre-fetching on the web is a technology to process user requests before they're actually needed. Therefore, the time that the user waits for the requested documents can be reduced by hiding latencies of the request. Prefetch is a way to reduce delays. The user always expects an interactive response, better satisfaction and output quality. Several methods and algorithms have been proposed to improve web performance.

Proposed Methodology: Web is a key resource in order to share the information along the world. It has large number of news, advertisements, global connectivity between people and lots of knowledge for the students. This massive use of Web or WWW makes it more important in the world of research. Researcher has the challenge to make the web applications more efficient. Many researchers work on it and give new idea in order to give the better results from the previous one. This dissertation is also puts its best foot forward in this era [4].
There is a huge need to improve the response time of server for web applications. Current Web has a massive repository due to increase its use suddenly. It has to focus on both the quality and quantity of web contents. Even, when the speed of Internet has improved with the reduced costs, the traffic is getting heavier. The enormous information makes it difficult to find the relevant information quickly. This led to the effort to improve the speed, by reducing the latency, make the web more relevant and meaningfully connected. [5].
The Cache prefetching plays an important role in order to enhance the response time and make the application well-organized. The web prefetching is a technique in order to preprocess the user requests, before they are actually demanded. Therefore, the time that the user must wait for the requested documents can be reduced by hiding the request latencies. Prefetching is the method for reducing Latencies. The user always expects an interactive response, better satisfaction and quality of output. There are various approaches

and algorithms have been proposed for improving the web performance [6].

The proposed work has improved the hit ratio of the web page and expedites users visiting speed. Predictive Web prefetching refers to the mechanism of deducing the forth coming page accesses of a client based on its past accesses. In this work, demonstrate the frequent mining pattern which is obtained on the basis of input and on the basis of that caching and prefetching ratio is calculated. Thus this work presents a new idea for the interpretation of Web prefetching and web caching from the given usage items. The approach works on the basis is web mining with the combination of markov model. A markov model describes a possible events in which the probability of each events depends only on the state attained in the previous events.

Big Data: The amount of data generated every day in the explosion of the world. The growing volume of digital media and social media and the Internet of Things, nourishes going further. Data growth is amazing, this data comes quickly, with a variety (and not necessarily regulated) and contains a wealth of information which can be a key to getting an edge in competing companies. "Big Data is a collection of very large data sets and complex that it becomes difficult to treat using traditional management databases or processing tools application data. The challenges in the areas of capturing, preservation, storage, search, sharing, transfer and analysis, and visualize these [7] data. “The world has been immersed in a sea of data today. In a wide range of application areas, data is collected on a scale never seen before. Decisions based on the above assumptions, or the carefully constructed reality models, can now be done on the basis of the same data. The analysis of large data now covering almost all aspects of modern society, including mobile services, retail, manufacturing, financial services, life sciences and physical sciences.

Big data is a term that describes the evolution of any structure, semi-structured and unstructured information that has the ability to operate for a sufficient amount of data information.

proposed work: The popularity of the World Wide Web has been increased in recent years. The charge has been observed in large number of Internet visitors. Some One could consider the World Wide Web for large distributed information systems to provide access to shared data. This work has been done to improve the responsiveness of the system on the Internet. It is also for the time of the distribution of information on the geographical location. Caching on the approaches of the Internet and significant preload used to significantly reduce the response time seen by users. Caching Ideal plan before putting the system is able to predict the next following applications and pre- load those caches are stored. The objects have been preloaded in the local cache to reduce memory latency.

Proposed Architecture of Analysis End User Behavior Over Web Scenario For Efficient Prefetching Through Big Data Analysis (AWPB): The proposed architecture AWPB (Analysis End User Behavior over Web Scenario for efficientPrefetching through Big Data) has three major things that are server, preprocessing of log file and transaction probability matrix of the pre-processing rules. First of all, the client requests to server as a Client Request. The server will process this request. It will analyze the prefetching page. This analysis will perform on the basis of the previous activity or requests of clients. Client gets reply as a Server Response. This reply will contain the prefetched page with pre analysis. This figure 2 shows the architecture of proposed work. The architecture of proposed work have the several components. In this proposed work there is a use of web server log in order to analysis the page which has been used to send with the reply of the client

Algorithm of AWPB:

Assumption

{

// P=Size of Weblog file in term of line

// R= {r1,r2,r3,,,,,,,rp} row id of log file over size P

// D= Dimension of Weblog file

// At = { A1,A2,A3,,,,,,,,,,Ad} attribute of log file over dimension D

// T= Token in Weblog file

Step 1 Token Reorganization and session identification

A. For every R over P identify each A_t as token

A_t^{ri} = phase of logfile between two separtor (:,/ \, –, [,]) in each r

B. For every Token over P identify session

if(A_t^{ri} = IP_t^{ri} ∉ (distinct session))

Add A_t^{ri} in set (distinct Session)

Else if(A_t^{ri} = (IP &OS)$_t^{ri}$ ∉ (distinct session))

Add A_t^{ri} in set (distinct Session)

Else if(A_t^{ri} = (IP , OS &browwser)$_t^{ri}$ ∉ (distinct Session))

Add A_t^{ri} in set (distinct Session)

Else if(A_t^{ri} = (IP , OS , browwser &referel uri)$_t^{ri}$ ∉ (distinct Session))

Add A_t^{ri} in set (distinct Session)

Step 3:- Link Prediction

Evaluate the inter session gap between each uri and the uri having lower session gap wrt other uri is call for pre fetching.

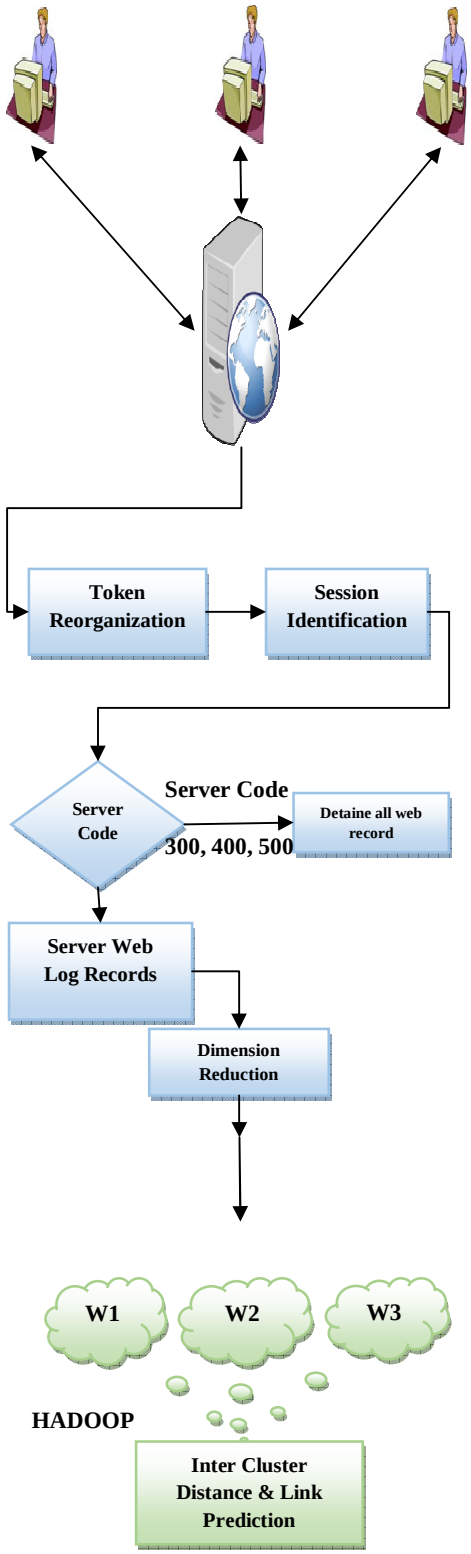

Figure 2 Architecture of AWPB

Components of AWPB

There are three major components used in this work. These are disused below.

- **Client:** In a web environment, there are two distinct computers or may be other devices, a client is any machine that requests for particular information, and the server is also a computer or machine which fulfills the client's requests. A web client is actually web browser that makes the requests to the remote server. Typically, a client is a computer application, such as a web browser, for example Internet Explorer (IE), Mozilla Firefox, Google Chrome etc., that runs on an end user's local computer or workstation and connects to a server. In server client architecture when they communicate they use HTTP or HTTPS and FTP protocol for exchanging information.
- **Web Server on Server Machine:** A web server is software program that performs operations in a client–server relationship in computer networking or it can be run on the same machine on which client is. Typically, a web server gets run on a remote machine, reachable from a user's local computer or workstation. The most common use of web servers is to host websites. Operations are performed on server machine because clients require access to information or functionality that is not available on the client.

Server-side operations also include processing and storage of data from a client to a server, which can be viewed by a group of clients. This lightens the work of your client. Examples of server-side processing include the creation & adaptation of a database. The primary function of a web server is to deliver web pages on the request to clients using the Hypertext

Transfer Protocol (HTTP). This means delivery of HTML documents and any additional content that may be included by a document, such as images, style sheets and scripts.

- **Web Server Log:** A file having ability to save "hit" of a Web site on server machine. This server machine is known as the web server, including each view of a HTML document, image or other object, is logged. The raw web log file format is essentially one line of text for each hit to the web site. This contains information about who was visiting the site, where they came from, and exactly what they were doing on the web site. In order to effectively manage a web server, it is necessary to get feedback about the activity and performance of the server as well as any problems that may be occurring. The Web Server provides very comprehensive and flexible logging capabilities in the form of log files.

Flow Chart : This data flow diagram shows that how does the client request to the server in order to access the web application. First of all client will send the request to the web server for their web application.

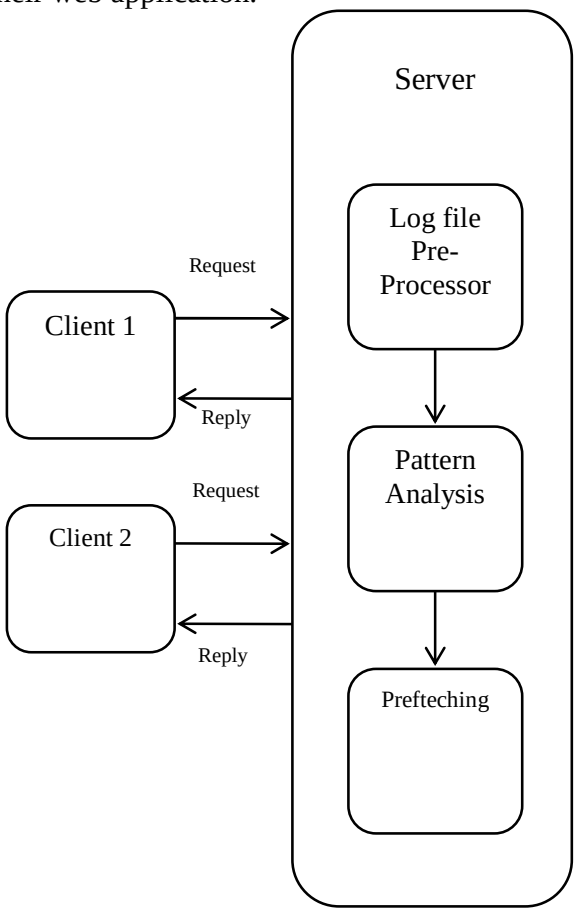

Figure 3 A Simple flow Diagram of AWPB

This is a simplified diagram in order to show the flow of request and reply in client server architecture. It shows how web server will forward the request to the application server and application server generates the reply. This reply also includes the prefetch page as per the proposed methodology. As figure 3 shows in server site there are three major steps will

follow. These three steps are log file pre-processor, pattern analysis and transaction matrix. These steps are used to calculate the page need to prefetch. When calculated page found it will send to Client by the reply of the web request.

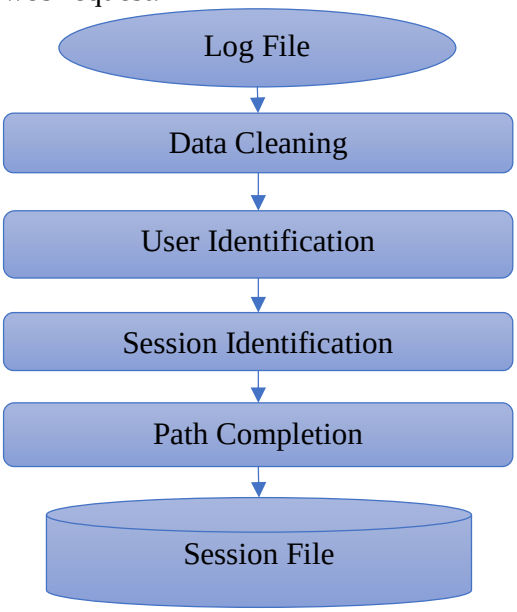

Figure 4 Log Pre-processing

Result Evaluation: This section show the comparative study of proposed work with existing technique .In proposed methodology number of Rules are required for finding the perfected page is relatively low as compare to existing technique with high hit ratio and lower access time.

Hit Ratio: Hit ratio is one of most common evaluation parameter to evaluate the performance of web prefetching technique that represent fraction of access that hit from cache to miss as shown in equation 1.1. For any ideal web page prefetching technique it is required to be higher hit ratio.

Table 1: Hit Ratio

Cache Hit	WSCRP	AWPB
10%	0.127	0.241
20%	0.276	0.347
30%	0.387	0.435
40%	0.505	0.6
50%	0.63	0.74
60%	0.704	0.76
70%	0.786	0.81
80%	0.872	0.89
90%	0.956	0.965
100%	0.994	0.995

$$Hit\ ratio = \frac{Access\ Hit}{Total\ Number\ of\ Access} \quad \ldots\ldots\ldots 1.1$$

Proposed Technique AWPB (Analysis End user Behavior over web Scenario for efficient Prefetching through Big Data) have higher hit ratio as compare to existing technique WSCRP as shown in table 1 and figure 5.

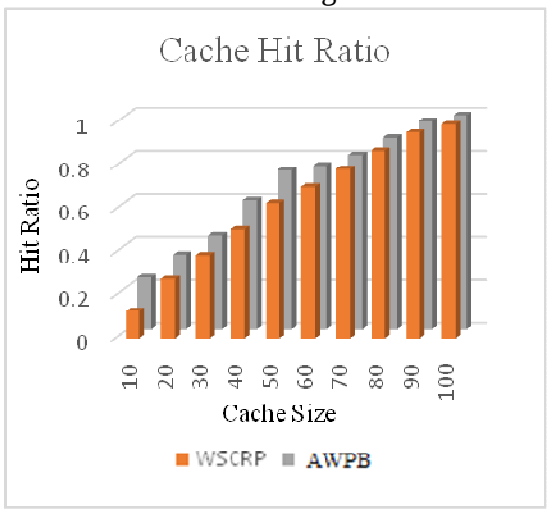

Figure 5: Cache Hit Ratio Comparison

Average Access Time: Average Access Time is another most common evaluation parameter to evaluate the performance of web prefetching technique that represent fraction of time need to access requested page and its depend upon hit ratio higher hit ratio having lower total access time. For any ideal web page prefetchingg technique it is required to be lower average access time

.

Table 2: Average Access Time

Cache Size	WSCRP	AWPB
10%	2736	2656
20%	2502	2475
30%	2427	2389
40%	2294	2234
50%	2106	2076

60%	1884	1834
70%	1706	1689
80%	1513	1490
90%	1428	1356
100%	1465	1340

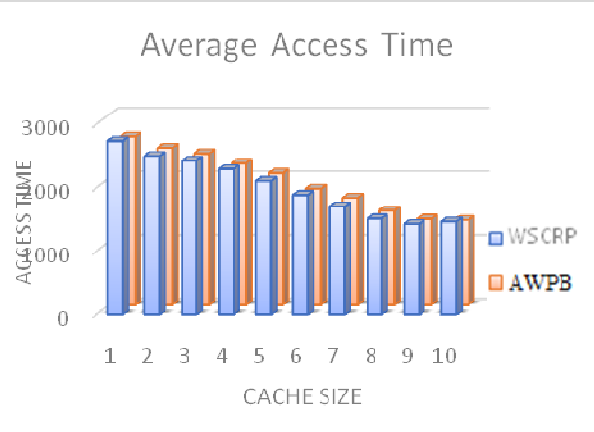

Figure 6.10 Average Access Time Comparisons

Proposed Technique AWPB (Analysis End User Behavior over Web Scenario for efficient Prefetching through Big Data Analysis) have lower average access time
as compare to existing technique WSCRP as shown in table 2 and figure 6.

Conclusion: The online prediction can go a long way in improving the user experience on the Internet. Web and Internet technology continues to evolve, and the ability to integrate these technologies is wide open. Ensure that the use of these techniques do not interfere indirectly on system performance must be supported. In this work, the Clustering technique and big data concept to work together in order to pre-fetch the page from the Web server.

Tests conducted in this work have proposed using AWPB based prefetching technique show that it performs better compared to previous work. It also shows the effect that it is easy to apply to preload the pages of the website.

References

1. K.Ramu, Dr.R.Sugumar and B.Shanmugasundaram "A Study on Web Prefetching Techniques" Journal of Advances in Computational Research: An International Journal Vol. 1 No. 1-2, January, 2012
2. Waleed Ali, Siti Mariyam Shamsuddin, and Abdul Samad Ismail "A Survey of Web Caching and Prefetching", Int. J. Advance. Soft Comput. Appl., Vol. 3, No. 1, March 2011
3. Daesung Lee and Kuinam J. Kim, "A Study on Improving Web Cache Server Performance Using Delayed Caching", IEEE 2010, pp 1-5.
4. A. Kala Karun and K. Chitharanjan, "A review on hadoop — HDFS infrastructure extensions," Information & Communication Technologies (ICT), 2013 IEEE Conference on, JeJu Island, 2013, pp. 132-137.
5. T. Hofmann. "The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data" In Proceedings of 16th International Joint Conference on Articial Intelligence IJCAI-99, pages 682-687, 1999.
6. T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "Websom-self-organizing maps of document collections", In Proc. of Workshop on Self-Organizing Maps (WSOM'97), pages 310-315, 1997
7. A. Saldhi, D. Yadav, D. Saksena, A. Goel, A. Saldhi and S. Indu, "Big data analysis using Hadoop cluster," Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on, Coimbatore, 2014, pp. 1-6.
8. B. D. Davison, "A Web caching primer," in IEEE Internet Computing, vol. 5, no. 4, pp. 38-45, Jul/Aug 2001.
9. G. Barish and K. Obraczke, "World Wide Web caching: trends and techniques," in IEEE Communications Magazine, vol. 38, no. 5, pp. 178-184, May 2000.
10. Pablo Rodriguez, Christian Spanner, and Ernst W. Biersack, "Analysis of Web Caching Architectures: Hierarchical and Distributed Caching", IEEE/ACM Transactions On Networking, Vol. 9, No. 4, August 2001
11. L. Ramaswamy, A. lyengar, L. Liu, F. Douglis, "Automatic Fragment Detection in Dynamic Web Pages and Its Impact on Caching", IEEE Transactions On

Knowledge And Data Engineering, Vol. 17, No. 6, June 2005.

12. P. Kolari and A. Joshi, "Web mining: Research and practice", Computer Science Engineering .July/August (2004) 42–53

13. B. Liu and K. Chang, "Editorial: Special issue on web content mining", SIGKDD Explorations 6(2) 2004, pp 1–4.

14. R. Kosala and H. Blockheel, "Web Mining Research: A Survey", In SIGKDD Explorations, Volume 2, Number 1, pages 1-15, 2000.