Volume 4, Issue 2, February 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Novel Approach for Web Usage Mining with Clicking Pattern in Grid Computing Environment

Archana Godbole
M.Tech (CTA) Research Scholar
RGTU University Bhopal, India

Mahendra Kumar Rai
M.Tech(IT) HOD
RGTU University Bhopal, India

Abstract— An Important task of web usage mining is to analyse users browsing sequences. It can help the Web administrators or designers to improve the web structure or tune the performance of the web servers. Web mining can be defined to find out the useful information form www. The analysis of Web log files may give information that is useful for improving the services offered by the users. MSCP is usually a costly task due to its considerable amount of time for computation and storage for archiving a huge amount of data. MSCP is inefficient and not workable on a computer with some resources. This paper explores the deployment of clicking pattern algorithms in a distributed grid computing environment and demonstrates its effectiveness by empirical cases. The basic task of this paper is a Web-log analysis for huge, widely distributed, hypertext information repository of www. So, the Web sites easily and automatically improve and tune up their organization. We propose latest clicking pattern algorithm SPMAST to discover Web usage patterns to analyse the visitor trends and compared this new approach from existing usage mining algorithms like GSP, SPAM and PrefixSpan algorithm in term of Storage and time efficiency in Advanced Distributive Grid Environment. SPMAST provides better accuracy as compared with GSP, SPAM and Prefixspan algorithm the use of advanced grid-environment and associated experimental results are presented and discussed.

Keywords— Web Usage Mining, Clicking Pattern, Gridsim, Grid Environment, SPMAST.

I. INTRODUCTION

A. Introduction to web usage mining

With the popularity of the Internet, to understand the intentions of the Internet users is of vital importance. Web usage mining [7, 6] is part of the tasks trying to understand the user's behaviours by analyzing the web logs. Its very purpose is to discover useful, but originally hidden, information or knowledge from massive web logs. Because web log data are increasing in sizes (number of transactions), one of the demands in web usage mining is the scalability, i.e. the ability to handle mashed data. Traditional algorithms are difficultIn scalability, due to their high algorithmic complexity and the limited capability to handle huge datasets. Therefore, high-performance parallel or distributed computation provides a feasible solution to this problem. It may reflect the common intentions or behaviours of the users when they visit the web sites. MSCP can help the web administrators or designers improve the web structure, tune the performance of the web servers, provide personalized services, etc However, MSCP is also a time-consuming task which usually needs to explore a huge search space exhaustively for forming meaningful clicking patterns. Recently, Distributive grid computing [2] is rapidly emerging as the dominant paradigm for large-scale parallel and distributed computing. Grid computing is the integrated use of heterogeneous or homogeneous computing resources, including computers, networks, And data storages, which may be geographically distributed, in order to create a virtual computing environment for solving large-scale problems. To improve the performance of MSCP, we present in this paper the deployment of latest clicking pattern mining algorithms executable in a distributive grid computing environment. The performance of MSCP can achieve in a comparatively easier way. The use of a Distributive grid environment and associated experimental results are presented and discussed. The Web Usage Mining use the data storage in the Log files of Web server as first resource; in this file the Web server register the access at each resource in the server by the users [5,6]. Generally, three kinds of information have to be handled in a web site: content, structure and log data. Content data contains of anything is in a web page, structure data is nothing but the organization of the content and usage data is nothing but the usage patterns of web sites. The usage of the data mining process to these dissimilar data sets is based on the three different research directions in the area of web mining: web content, structure and usage mining. Web usage mining is the type of data mining process for discovering the usage patterns from web information for the purpose of understanding and better provide the requirements of web-based applications. Web usage mining imitates the performances of humans as they interact with the Internet. Examination of user actions in communication with web site can offer insights causing to customization and personalization of a user's web practice. As a result of this, web usage mining is of extremely useful for e-marketing and ecommerce professionals. Web usage mining involves of Three phases like pre-processing, pattern discovery and pattern analysis. There are different techniques available for web usage mining with its own advantages and disadvantages. This paper provides some discussion about some of the techniques available for web usage mining. In this paper, we propose latest clicking pattern algorithms to discover Web usage patterns (data clusters) to analyze the visitor trends and

compared to Traditional sequential clicking pattern based algorithm in term of storage and time efficiency. Sequential pattern based SPAM algorithm provides better accuracy compared with GSP, Prifixspan and Apriori algorithms.

II. CLICKING PATTERN GENERATION

Many algorithms are used to find out frequent sequential clicking patterns from the web log. It helps users to understand the natural grouping or structure in a data set. We analyze some latest clicking pattern algorithms implemented in distributive grid environment and discussed experimental results of each algorithms.

A. Algorithm GSP

The algorithm makes multiple passes over the data. The first pass determines the support of each item, that is, the number of data-sequences that include the item. At the end of the first pass, the algorithm knows which items are frequent, that is, have minimum support. Each such item yields a 1-element frequent sequence consisting of that item. Each subsequent pass starts with a seed set: the frequent sequences found in the previous pass. The seed set is used to generate new potentially frequent sequences, called candidate sequences. Each candidate sequence has one more item than a seed sequence; so all the candidate sequences in a pass will have the same number of items. The support for these candidate sequences is found during the pass over the data. At the end of the pass, the algorithm determines which of the candidate sequences are actually frequent. These frequent candidates become the seed for the next pass. The algorithm terminates when there are no frequent sequences at the end of a pass, or when there are no candidate sequences generated.[4] .

B. The Prefix span algorithm

Prefix Span comes under pattern growth method for mining sequential patterns. Its major idea is that, instead of projecting sequence databases by considering all the possible occurrences of frequent subsequence's, the projection is based only on frequent prefixes because any frequent subsequence can always be found by growing a

Frequent prefix. Prefix Span examines only the prefix subsequences and projects only their corresponding postfix subsequence's into projected databases. This way, sequential patterns are grown in each projected database by exploring only local frequent sequences. Steps of the algorithm are:

Step 1: Scan the database once

Step 2: Get the frequent item from the database such that the occurrence of *fi* should be equal to or greater than minimum support

Step 3: For i= 0 to list of all frequent items

i) On basis of frequent item list get the suffix subsequences from database of that supplied $f_i[i]$.

ii) Get the frequent item from the subsequence and append to the List.

Step 4: Repeat Step 3 till end of all frequent items.

C. The SPAM Algorithm

The SPAM algorithm is based upon lexicographic tree of sequence. In this algorithm the tree traversal and the pruning methods are used to reduce search space. The algorithm is especially efficient when the sequential patterns in the database are very long. A salient feature of our algorithm is that it incrementally outputs new frequent itemsets in an online fashion. This algorithm specially works in two ways described below. Steps of the algorithm are as follows:

1. SPAM Traverses candidate sequences as defined by a lexicographic sequence Tree.
2. Each sequence in this tree can be considered as either a sequence-extend sequence or as an itemset-extended sequence.
3. A Sequence-extended sequence is generated by adding a new transaction consisting of a single item to the end of its parent's sequence in the tree.
4. An itemset-extended sequence is generated by adding an item to the last itemset in the parent's sequence, such that the item is greater than any item in that last itemset.
5. The transitions from a parent to sequence extended and itemset-extended sequences as S-steps and I-steps respectively.
6. A lexicographic sequence tree with the restriction that no sequence may contain more than two item depicted in the figure 1

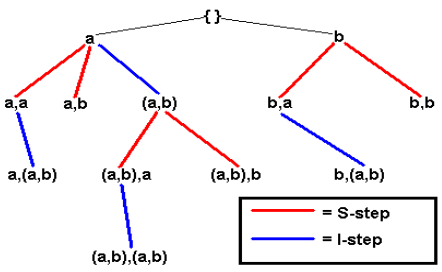

Fig 1

7. SPAM uses both S-step and I-step pruning to reduce the search space.
8. Suppose we are currently at the node for sequence ({a}) and can reach sequences ({a},{a}),({a},{b}),({a},{c}) by S-steps. If it turns out that ({a},{c}) is not frequent, we know by the Apriori principle that ({a},{a},{c}),({a},{b},{c}),({a},{a,c}), and ({a},{b,c}) all cannot be frequent. This is called pruning mechanism S-step pruning.

9. Now we can reach ({a,b}) and ({a,c}) via I-steps from ({a}). If ({a,c}) is not frequent, than ({a,b,c}) must also not be frequent by the Apriori principle this term as I-step pruning.

D. SPMAST Algorithm

Proposed algorithm (SPMAST) works on a projection-based incremental mining algorithm of sequential patterns. Its main idea is to convert all the sequences in the incremental database into the new insert sequences that are not associated with the sequences in the original database, construct the projected database for the new insert sequences and complete the updated operations of the sequence tree. First of all, the sequences in the incrementaldatabase are divided into two parts, one is the set of insert sequences, and the other is the set of append sequences. The append sequences are the extension of the related sequences in the original database, so we need to find the sequences in the original database that have the same sequence ID with the append sequences. These sequences in the original database constitute the set of sequences, called App_Original. App_Original is combined with the set of append sequences to form a new set of append sequences. Then, we construct the projected databases for App_Original to find all the sequences and its support that associated with App_Original, and delete the sequences and its support from the sequence tree. All the sequences associated with App_Original have been removed from the sequence tree, so we can consider the sequences in the new set of append sequences as the insert sequences. The projected databases are constructed for the set of insert sequences and the new set of append sequences, and all the sequences and its support that generated in the process of construction are added to the sequence tree. At present, the sequence tree contains all the sequences and its support in the updated database, and we can find all the sequential patterns that meet support threshold through traversing the sequence tree So the steps of the algorithm are as follows:

SPMAST (D, db, min_sup, Stree)
Input: An original database D, an incremental database db, minimum support threshold min_sup, sequence tree Stree
Output: The updated sequence tree Stree, The updated database D, FS' in the updated database
1: If db ≠ NULL // Database is updated.
2: Scan the incremental database db, and find Insert_ Set and App_Set;
3: Scan the original database D, and find App_Original;
4: Construct the projected database for App_Original, where minimum support threshold is 1, and get FS_ App_Original;
5: Delete the sequences and its support in FS_ App_Original from the sequences tree;
6: App_Updated = App_Original + App_Set;
7: db' = Insert_Set + App_Updated;
8: Construct the projected database for db', where minimum support threshold is 1, and get FS_db';
9: Add the sequences and its support in FS_ db' to the sequences tree, and get the updated sequence tree Stree;
10: Traverse_Stree(root, min_sup, Stree);
11: D=D - App_Original + db';
// Update the original database D.
12: ElseIf db = NULL // The support is changed.
13: Traverse_Stree(root, min_sup, Stree);
14: **Return;**
SPMAST_ Traverse_Stree(α, min_sup, Stree)
Input: The root node α, minimum support threshold min_sup, sequence tree Stree
Output: The set of frequent patterns FS'
1: If Stree ≠ NULL
2: Scan the sequence tree, and find all the child nodes of the root node α, called child_node;
3: For each node s in child_node do
4: If support(s) ≥ min_support
5: FS'= FS' + s;
6: Traverse_Stree(s, min_sup, Stree);
7: **Return;**

III. TASK AND DATA DECOMPOSITION

In the present work, this paper proposes a novel approach of latest clicking pattern mining algorithm to identify the user's habits in distributive gird environment. This kind of clicking pattern approach will be tried to gather the users by patterns

of pages accesses. To obtain this result we need to process the Web Log files to identify users and session of users in distributive gird environment.

IV. DISTRIBUTIVE GRID ENVIEONMENT

In this paper we present a high-level overview of distributive grid computing. The proliferation of the Internet and the availability of powerful computers and high-speed networks as low-cost commodity components are Changing the way we do large-scale parallel and distributed computing. The interest in coupling geographically distributed (computational) resources is also growing for solving big problems, leading to what is popularly called the grid and peer-to-peer (P2P) computing networks. These enable sharing, selection and aggregation of suitable computational and data resources for solving large-scale data intensive problems in science, engineering, and commerce. A generic view of grid computing environment is shown in Figure 1 the grid consists of four key layers of components: fabric, core middleware, user-level middleware, and applications. The grid fabric includes computers (low-end and high-end computers including clusters), networks, scientific instruments, and their resource management systems. The core grid middleware provides services that are essential for securely accessing remote resources uniformly and transparently. The services they provide include security and access management, remote job submission, storage, and resource information. The user-level middleware provides higher-level tools such as resource brokers, application development and adaptive runtime environment.

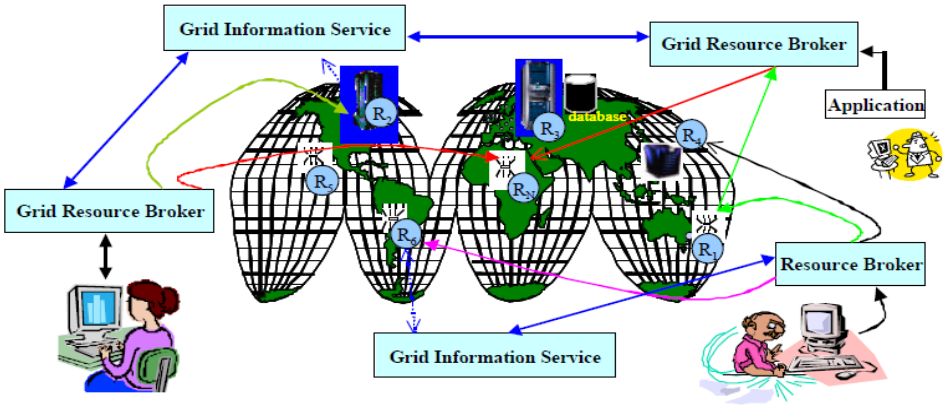

Fig 2: Grid Environment

V. EXPEREMENTS AND RESULTS

For simulation of grid environment we used Gridsim toolkit. Eclipse is used as the IDE for running Gridsim toolkit. GridSim provides a simulation environment for the Grid computation. We implemented the algorithms discussed in the previous section and analyses the performance of each algorithm on the basis of some evaluation parameters. The snapshot of the SPMAST algorithm in the Grid environment is shown below:

Fig 3 Snapshot of SPAMST in Grid Environment

Data Set Used

Real-life dataset used in order to evaluate the performance of the algorithm, which is as follows:

a)* MSNBC*: a dataset of click-stream data. The original dataset contains 989,818 sequences obtained from the UCI repository. Here the shortest sequences have been removed to keep only 31,790 sequences. The number of distinct item in this dataset is 17 (an item is a webpage category). The average number of itemsets per sequence is13.33. The average number of distinct item per sequence is 5.33.

The proposed algorithm (SPMAST) and the previous work have been analyzed on the basis of the some evaluation criteria like Time elapsed in execution, Storage occupancy and frequent sequence count. In this section empirical and functional behaviors of the algorithm have been analyzed for MSNBC dataset. The comparative analysis of SPMAST with existing algorithms on MSNBC dataset is shown in the depicted table:

Table 1 Comparison with Minsup 0.2 (with MSNBC dataset)

Algorithm 0.2	Max. Pattern Length	Time(in ms)	Max Memory (in mb)	Frequent Sequences count
SPMAST	4	630	116	68
GSP	4	11960	152.3	73
SPAM	4	2280	159.6	67
PREFIX SPAN	4	23890	227.2	67

we can derive the conclusion that behavior of SPMAST, SPAM algorithms are similar for the frequent sequence count. The SPMAST algorithm is superior to the rest of algorithms, while execution time and storage occupancy is considered. GSP and Prefixspan algorithm takes significantly more time than the SPAM and SPMAST (Proposed algorithm).As far as frequent sequence count is concern the SPMAST algorithm performs outstanding and it is also exhibit good performance for the rest of performance criteria as it scans the database once.

Table 2. Comparison with Minsup 0.3(with MSNBC dataset)

Algorithm 0.3	Max. Pattern Length	Time(in ms)	Max Memory (in mb)	Frequent Sequences count
SPMAST	4	404	142.8	26
GSP	4	6029	211.9	27
SPAM	4	1570	71	26
PREFIX SPAN	4	12216	218.2	26

With minimum support threshold 0.3 (30%) the empirical data for the proposed and the rest of web usage mining algorithms are shown in Table 2 .The SPMAST algorithm is again performs better than the rest of algorithms, while execution time and storage occupancy is considered. GSP and Prefix SPAN algorithm takes significantly more time than the SPMAST and SPAM.
The graphical analysis shows the execution time of all algorithms as follows:

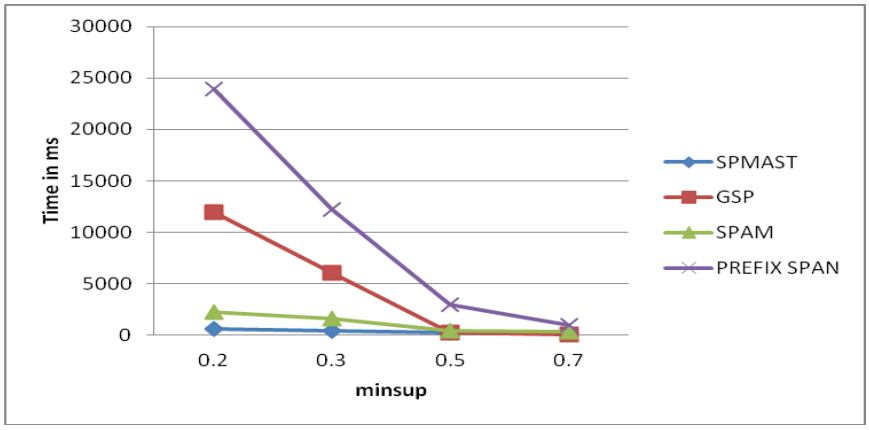

Fig 4. Analysis of Execution time

VI. Conclusion

In this work, we study the possible use of the clicking pattern mining algorithms to classify the web traffic data mining set in discovery of useful knowledge. We can conclude that; to identify common patterns in Web, SPMAST Algorithm is better than GSP, SPAM and prifixspan. Mining sequential clicking pattern (MSCP) is not well suited for grid environments because MSCP is a time-consuming task which usually needs to explore a huge search space exhaustively for forming meaningful clicking patterns. To improve the performance of web usage mining in grid environment we proposed a new algorithm SPMAST and compare its performance with existing algorithms of mining namely Prefix-span, SPAM and GSP. Among all of the algorithms SPMAST algorithm takes less memory space for execution. The number of algorithms is very large as far as the field of web mining is considered and it is hard to check and compare the performance of each and every algorithm. In future all Web Usage Mining algorithms can be implemented for cloud, big data and HPC environment to achieve better computation results. Also the use of different dataset can make a major impact on the results, so new datasets can also be used in cloud, HPC and big data environment.

References

[1] Nina, S.P., Rahman, M., Bhuiyan, K.I. and Ahmed, K., "Pattern Discovery of Web Usage Mining", International Conference on Computer Technology and Development, Vol. 1, Pp.499-503, 2009.

[2] Chih-Hung Wu, Yen-Liang Wu, Yuan-Ming Chang and Ming-Hung Hung, "Web Usage Mining on the Sequences of Clicking Patterns in a Grid Computing Environment", International Conference on Machine Learning and Cybernetics (ICMLC), Vol. 6, Pp. 2909-2914, 2010

[3] Aghabozorgi, S.R. and Wah, T.Y., "Using Incremental Fuzzy Clustering to Web Usage Mining", International Conference of Soft Computing and Pattern Recognition, Pp. 653-658, 2009.

[4] Rahul Neve , K.P Adhiya," Comparative Study of Web Mining Algorithms For Web Page Prediction In Recommendation System**"** , International Journal of advanced Research in computer and communication Engineering, Volume 2, Pp 969-976 , 2013.

[5] Jalali, M., Mustapha, N., Sulaiman, N.B. and Mamat, A., "A Web Usage Mining Approach Based on LCS Algorithm in Online Predicting Recommendation Systems", 12th International Conference Information Visualisation, Pp. 302-307, 2008.

