ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

*Website: www.ijircce.com*

**Vol. 5, Issue 1, January 2017**

# A Survey of Various Load Balancing Techniques

Suraj Patil, Prof. Parth Sagar

PG Student, Dept. of Computer Engg. RMD Sinhagd School of Engineering Warje, Pune, India

Dept. of Computer Engg, RMD Sinhagd School of Engineering Warje, Pune, India

**ABSTRACT:** Frequent sequence mining is well known data mining. The output of the algorithm is used in many other areas like chemistry, bioinformatics, and market basket analysis. Unfortunately, the frequent sequence mining is computationally quite expensive. we present a novel parallel algorithm for mining of frequent sequences based on a static load-balancing. The static load balancing is done by measuring the computational time using a probabilistic algorithm and parallel algorithm. For reasonable size of instance, the algorithms achieve speedups up to =3/4 P where P is the number of processors. In the experimental evaluation, our method performs significantly better than the current state-of-the-art methods. The presented approach is very universal it can be used for static load balancing of other pattern mining and parallel mining algorithms such as item set/tree/graph mining algorithms.

**KEYWORDS:** Data mining, frequent sequence mining, parallel algorithms, static load-balancing, probabilistic algorithms.

## I. INTRODUCTION

Repeated pattern removal is an important data mining technique with a wide variety of mined patterns. The mined frequent patterns can be sets of items (item sets), sequences, graphs, trees, etc. Frequent sequence mining was first described in. The GSP algorithm presented in is the first to solve the problem of frequent sequence mining.
As the repeated series removal is an extension of item set mining, the GSP algorithm is an extension of the Apriori algorithm. As a consequence of the slowness and memory consumption of algorithms described in other algorithms were proposed. These two algorithms use the so-called prefix-based equivalence classes (PBECs in short), i.e., represent the pattern as a string and partition the set of all patterns into disjoint sets using prefixes. There are two kinds of parallel computers: shared memory technology and distributed memory technology.
Parallelizing on the shared memory technology is easier than parallelizing on distributed memory technology. Sampling technique that statically load-balance the computation of parallel frequent item set mining process, are proposed in these three papers, the so-called double sampling process and its three variants were proposed.

## II. LITRATURE SURVEY

In the Load balancing important things are estimation of load, comparison of load, stability of different system, performance of system, interaction between the data sets, nature of work to be transferred, selecting of data sets and many other ones to consider while developing such algorithm
Samplingtechnique that statically load-balance the computation of parallel frequent itemset mining process, are proposed in [2] The double sampling process is enhanced by introducing weights that represents the relative processing time of the algorithm for a particular PBEC. Other algorithms were proposed. Thetwo major ideas in the frequent sequence mining are those of Zakiand Pei and Han[13]. These two algorithms use the so-called prefixbased equivalence classes (PBECsinshort), i.e., represent the pattern and partition the set of all patterns into disjoint sets using prefixes. The two algorithms [11], [5] differ only in the datastructures usedtocontrolthesearch. the sequential algorithm runs for too long there is a need for parallel algorithms. Such as the one described in this paper. There is a very natural opportunity to parallelize an arbitrary frequent sequence mining algorithm: partition

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

thesetofallfrequentsequencesusingthePBECs. The GSP algorithm presented in [7] is the first to solve the problem of frequent sequence mining. As the frequent sequence mining is an extension of itemset mining, the GSP algorithm is an extension of the Apriori algorithm [3].
The Apriori and the GSP algorithms are breadth first search algorithms. The GSP algorithm suffers with similar problems as the Apriori algorithm: it is slow and memory consuming. Freespan algorithm [12] is an example of one of the first DFS algorithms. The algorithm was enhanced in the Prefix- span algorithm [9], which uses the pseudo-projected database format, introduced for frequent itemset mining in [13]. The pseudo-projected database is in fact very similar to the vertical representation of the database used in the Spade algorithm [5]. Our method uses the Prefix span algorithm and its operations as a base sequential algorithm. There is also an algorithm that extends the tree projection algorithm for mining of frequent items to sequences [10].

## III. PROPOSED APPROACH

**A. Problem Definition**: Main Issue is the Data Being Used as Dynamic item set, Secondly the BFS algorithm are used. The Apriori algorithm is a BFS algorithm initially created for mining of frequent item sets
**B. Proposed System Overview:** Proposed method is a novel parallel method that statically load balance the computation. The set of all frequent sequences is first split into PBECs, the relative execution time of each PBEC is estimated and finally the PBECs algorithm is assigned to processors. The method estimates the processing time of one PBEC by the sequential Prefix span algorithm using sampling data sets. It is important to be aware that the running time of the parallel sequential algorithm scales with points 1) the database size 2) the number of frequent sequences 3) the number of embedding of a frequent sequence in database transactions.

## IV. A COMPARATIVE STUDY OF SEQUENTIAL PATTERN MINING ALGORITHMS

Depending on the management of the corresponding sequential database, andsequential pattern mining can be divided into three categories of databases, namely: a) Static database, b) Incremental database and c) Progressive database. Table 1 gives a comparison of the previously described algorithms based on the following features:
Database Multi-Scan: This feature includes the original database scanning to discover whether a long list of produced candidate sequences is frequent or not.
Candidate Sequence Pruning: This feature allows some algorithms (Pattern-growth algorithms, and later early-pruning algorithms) to utilize a data structure allowing them to prune candidate sequences early in the mining process.
Search Space Partitioning: This feature is a characteristic feature of pattern-growth algorithms. It permits the partitioning of the generated search space of large candidate sequences for efficient memory management.
DFS based approach: With the use of DFS search approach, all sub-arrangements on a path must be explored before moving to the next one.
BFS based approach: This feature allows level- by-level search to be conducted to find the complete set of patterns (All the children of a node are processed before moving to the next level)
Regular expression constraint: This feature has a good property called growth-based anti- monotonic. A constraint is growth-based anti- monotonic if it includes the following property: A sequence must be reachable by growing from any component which matches part of the regular expression, if it satisfies the constraint.
Top-down search: This feature has the following characteristic: the mining of sequential patterns subsets can be done by the corresponding set construction of projected databases and mining each recursively from top to bottom.
Bottom-up search: The Apriori-based approaches use a bottom-up search (from bottom to top), specifying every single frequent sequence.
Tree-projection: This feature allows simulating split-and-project by employing conditional search on the search space represented by a tree. It is used as an alternative in-memory database because it supports counting avoidance.
Suffix growth vs Prefix growth: This feature allows that the frequent subsequences exist by growing a frequent prefix/suffix; since it is usually common among a good number of these sequences. This characteristic reduces the amount of memory required to store all the different candidate sequences sharing the same prefix/suffix.
Database vertical projection: Algorithms using this feature visit the sequence database only once or twice to obtain a vertical layout of the database

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

Table 1: Comparative Study of some Sequential Pattern Mining Algorithms

| | Apriori All | GSP | SPADE | FreeSpan | PrefixSpan | ISM | SPAM | IncSp | ISE | IncSpan |
|---|---|---|---|---|---|---|---|---|---|---|
| Statical database | √ | √ | √ | √ | √ | N/A | √ | N/A | N/A | N/A |
| Incremental database | N/A | N/A | N/A | N/A | N/A | √ | N/A | √ | √ | √ |
| Candidate Sequence Pruning | N/A | √ | √ | N/A | √ | √ | N/A | √ | √ | √ |
| Search Space Partitioning | √ | N/A | N/A | N/A | N/A | N/A | N/A | √ | N/A | √ |
| DataBase MultiScan | √ | √ | N/A | N/A | N/A | N/A | N/A | N/A | √ | N/A |
| DFS based approach | N/A | N/A | √ | √ | √ | N/A | √ | N/A | N/A | N/A |
| BFS based approach | | √ | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| Regular expression constraint | N/A | N/A | N/A | √ | √ | N/A | N/A | N/A | N/A | N/A |
| Top-down search | N/A | N/A | N/A | √ | √ | N/A | N/A | N/A | N/A | N/A |
| Bottom-up search | N/A | √ | √ | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| Tree-projection | N/A | N/A | N/A | N/A | N/A | √ | N/A | N/A | N/A | N/A |
| Prefix growth | N/A | N/A | N/A | N/A | √ | N/A | N/A | N/A | √ | N/A |
| Database vertical projection | N/A | N/A | √ | N/A | N/A | N/A | √ | N/A | √ | N/A |

## V. CONCLUSION AND FUTURE WORK

We built up a ranking fraud recognition framework for mobile Apps. In particular, we initially demonstrated that ranking fraud happened in leading sessions and gave a technique to mining leading sessions for each App from its verifiable ranking records. At that point, we distinguished ranking based evidences, rating based proofs and survey based proofs for recognizing ranking fraud. Also, we proposed a optimization based accumulation strategy to coordinate every one of the evidences for assessing the validity of leading sessions from mobile Apps. A one of a kind point of view of this approach is that every one of the proofs can be demonstrated by factual speculation tests, subsequently it is anything but difficult to be reached out with different evidences from space learning to recognize ranking fraud. At last, we approve the proposed framework with broad tests on true App information gathered from the Google Play store.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

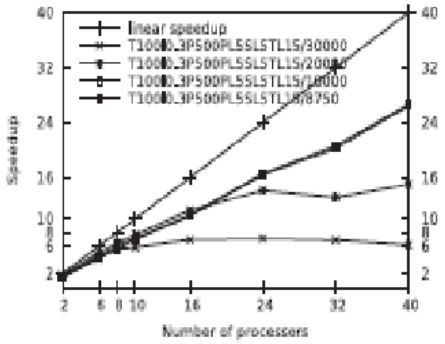
(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

## VI. RESULTS

The Proposed method is experimentally evaluated. The whole algorithm was implemented in C++ (compiled with gcc 4.4) using MPI, resulting in _ 30,000 lines of code. The implementation was executed on the CESNET metacentrum on the zegox cluster. Each zegox's node contains two Intel E5-2620 equipped with 1_-Infiniband. Nodes were exclusively allocated for these measurements and used a maximum of five cores per node (to avoid influences from other jobs).




## REFERENCES

1. T. Shintani and M. Kitsuregawa, Mining algorithms for sequential patterns in parallel: Hash based approach, in Proc. 2nd Pacific-Asia Conf., Res. Develop. Knowl. Discovery Data Mining, 1998, pp. 283294.
2. R. Kessl and P. Tvrd, Toward more parallel frequent itemset mining algorithms, in Proc. 19th IASTED Int. Conf. Parallel Distrib. Comput. Syst., 2007, pp. 97103.
3. J. Ren, Y. Dong, and H. He, A parallel algorithm based on prefix tree for sequence pattern mining, in Proc. 1st ACIS Int. Symp. Cryptography Netw. Security, Data Mining Knowl. Discovery, E-Commerce Appl. Embedded Syst., 2010, pp. 611.
4. R. Srikant and R. Agrawal, Mining sequential patterns Generalizations and performance improvements,in Proc. 5th Int. Conf. Extending Database Technol.: Adv. Database Technol., 1996, pp. 117.
5. R. Kessl and P. Tvrd, Probabilistic load balancing method for parallel mining of all frequent itemsets, in Proc. 18th IASTED Int. Conf. Parallel Distrib. Comput. Syst., 2006, pp. 578586.
6. V. Guralnik, N. Garg, and G. Karypis, Parallel tree projection algorithm for sequence mining,in Proc. 7th Int. Euro-Par Conf. Euro-Par Parallel Process., 2001, pp. 310320.
7. S. Cong, J. Han, J. Hoeflinger, and D. Padua, A samplingbased framework for parallel data mining, in Proc.10th ACM SIGPLAN Symp. Principles Practice Parallel Program., 2005, pp. 255265.
8. V. Guralnik and G. Karypis, Dynamic load balancing algorithms for sequence mining, Univ. Minnesota, Minneapolis, MN, US, Tech. Rep. TR 01-020, 2001.
9. S. Cong, J. Han, J. Hoeflinger, and D. Padua, A samplingbased framework for parallel data mining, in Proc.10th ACM SIGPLAN Symp. Principles Practice Parallel Program., 2005, pp. 255265.
10. M. J. Zaki, Parallel sequence mining on shared-memory machines, J. Parallel Distrib. Comput., vol. 61, no. 3, pp. 401426, 2001.
11. M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, New algorithms for fast discovery of association rules, in Proc. 3rd Int. Conf. Knowl. Discovery Data Mining, 1997, pp. 283286.
12. K.-M. Yu and J. Zhou, Parallel TID-based frequent pattern mining algorithm on a PC cluster and grid computing system, Expert Syst. Appl., vol. 37, no. 3, pp. 24862494, 2010.
13. M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, New algorithms for fast discovery of association rules, in Proc. 3rd Int. Conf. Knowl. Discovery Data Mining, 1997, pp. 283286.