Distributed Sequential Pattern Mining: A Survey and Future Scope

Suhasini Itkar
Assistant Professor
Department of Computer Engineering
Modern college of Engineering
Pune, India

Uday Kulkarni
Professor
Department of Computer Science & Engineering
SGGS Institute of Engineering and Technology
Nanded, India

ABSTRACT
Distributed sequential pattern mining is the data mining method to discover sequential patterns from large sequential database on distributed environment. It is used in many wide applications including web mining, customer shopping record, biomedical analysis, scientific research, etc. A large research has been done on sequential pattern mining on various distributed environments like Grid, Hadoop, Cluster, Cloud, etc. Different types of sequential pattern mining can be performed are sequential patterns, maximal sequential patterns, closed sequences, constraint based and time interval based sequential patterns. This paper presents a systematic review on work done for sequential pattern mining and advanced sequential pattern mining on distributed environment. This paper finally presents future research directions related to sequential pattern mining in distributed environment.

General Terms
Association Rule Mining, Sequential Pattern Mining.

Keywords
Distributed Sequential Pattern Mining, Maximal Patterns, Constraint based Patterns, Distributed environment.

1. INTRODUCTION
Sequential pattern mining is discovering sequential patterns from large sequence database. Sequential pattern can be widely used in customer purchase patterns for inventory control, web access patterns for websites, analysis of sequences or time related processes such as scientific experiments, natural disasters, disease treatment, analysis of DNA sequences, etc.

The problem of finding sequential pattern was first proposed in [1]. There are different approaches to mine sequential patterns like Apriori-based algorithm GSP [2], SPAM [3], projection-based FreeSpan [4], PSPM [5], vertical data format based algorithm SPADE [6] and pattern growth based approach in UDDAG [7] have been proposed. There are different specialized ways to find the sequential patterns which are mining of multidimensional association rules involve more than one dimension [8], mining of closed patterns [9] [10], maximal patterns [11], Constraint based mining [12] [13], approximate patterns [14]. Above mentioned algorithms are mainly executed on standalone environment which has some drawbacks like large scanning time for database, scalability problem, less efficient for massive dataset. To improve the performance of sequential pattern mining and to improve the scalability issues many researchers provide different techniques to work on distributed environment like grid computing, cluster, cloud, Hadoop, etc. and distribute the mining computation over more than one node.

The remaining paper is organized as follows. We define the theoretical foundations and related work of distributed sequential pattern mining in section 2. Taxonomy of various algorithms in distributed sequential pattern mining is mentioned in Section 3. Section 4 addresses comparative analysis of distributed sequential pattern mining algorithms. Section 5 conclude the study and explain some challenging issues for future scope.

2. THEORETICAL FOUNDATION AND RELATED WORK
This section represents the problem statement and attributes for sequential and distributed sequential pattern mining and related work done for sequential pattern mining in distributed environment.

2.1 Problem Statement
Let $I = \{i_1, i_2, \dots, i_k\}$ be a set of all items. A subset of I is called an itemset. A sequence $\alpha = \langle t_1, t_2, \dots, t_m \rangle (t_i \subseteq I)$ is an ordered list [3]. Each itemset in a sequence represents a set of events happening at the same timestamp, while different itemsets occur at different times. For example, a customer shopping sequence could be buying several products on one trip to the store and making several subsequent purchases, e.g., buying a PC then antivirus and some software, followed by buying a digital camera, memory card and a card reader, and finally buying a printer.

Without loss of generality, we assume that the items in each itemset are sorted in certain order (such as alphabetic order or ascending order).

Definition 1. Sequential Pattern Mining: A sequence $\alpha = \langle a_1, a_2, \dots, a_m \rangle$ is a sub-sequence of another sequence $\beta = \langle b_1, b_2, \dots, b_n \rangle$, denoted by $\alpha \sqsubseteq \beta$ (if $\alpha \neq \beta$, written as $\alpha \sqsubset \beta$), if and only if $\exists\, i_1, i_2, \dots, i_m$ such that $1 \leq i_1 < i_2, \dots < i_m \leq n$ and $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}, \dots$, and $a_m \subseteq b_{i_m}$. We also call β a supersequence of α, and β contains α. Given a sequence database $D = \{s_1, s_2, \dots, s_3\}$, the support of a sequence α is the number of sequence in D which contain α. If the support of a sequence α satisfies a pre-defined min $\_sup$ threshold, is a frequent sequential pattern.

Definition 2. Distributed Sequential Pattern Mining: Let $I = \{i_1, i_2, \dots, i_n\} = \{a, c, e, f, h, k, l, n\}$ be a set of all items. A sequence database DB is a set of tuples which contains SID sequence id and element sequence. The support or frequency $S\%$ of a sequence α in sequence database means S % of the

tuples in DB involving α and the sequence α is called sequential pattern. Suppose database DB is distributed horizontally and n nodes want to mine sequential patterns. That is, the database DB is horizontal distributed on n nodes $(s_1, \dots, s_n)$, and $DB = DB_1 \cup DB_2 \cup \dots \cup DB_n$ where DB_i is distributed on s_i . Let the global support threshold is S %. Let $X.sup$ and $X.sup_i$ be the frequency counts of an element X in DB and DB_i respectively. Then, the sum of global support of X in DB is computing as $X.sup = \sum_{i=1}^{n} X.sup_i$. The sequence X is globally large if $X.sup \geq s\ \% \times (\sum_{i=1}^{n} |DB_i|)$ and X is called global sequential pattern.

2.2 Related Work

Based on unity of patterns to be mined patterns are categorized as closed sequences, constraint based and maximal patterns which are discussed in closed frequent itemset in [15] [16], constraint frequent itemset in [17] [18] and maximal frequent itemset in [19][20].

Mining sequential patterns have some disadvantages like scalability problem; maximum time required for scanning the database, unable to store patterns in memory and is not suitable for large dataset mining. So moving towards distributed environment is very important to solve the challenges in sequential pattern mining. Sequential pattern mining can be implemented in distributed manner.

A variant of the parallel tree projection based frequent sequence mining algorithm is designed in [21]. The parallel closed sequential pattern mining algorithm (Par-CSP) which executes on distributed memory system was introduced in [22].

In [23] [24] Apriori based GSP algorithm is implemented on grid computing environment which is easy for loosely coupled methods. Grid computing environment requires a complicated method of data partitioning, approach to determine and assign the job to grid node dynamically and less powerful grid can degrade the performance of complete grid system.

Cluster based algorithm is developed in [25], where based on similarity definition algorithm groups the sequence data into some clusters and then distribute the cluster on distributed memory parallel computer nodes.

Standing for Sequential Pattern Mining on the Cloud (SPAMC) algorithm is implemented in [26] which are based on MapReduce framework and it is extended from SPAM. Sequential pattern mining on massive dataset on Hadoop environment has been proposed in [27] which is based on MapReduce programming model and uses PrefixSpan algorithm [28]. Distributed sites are used for sequential pattern mining of multidimensional sequential patterns in [29].

3. TAXONOMY OF DISTRIBUTED SEQUENTIAL PATTERN MINING

This section explains sequential pattern mining algorithms which have been implemented on distributed environment. Distributed sequential pattern mining is categorized in two parts as basic sequential pattern mining on distributed environment and advanced distributed sequential pattern mining with specific types like maximal, approximate and time interval based patterns.

3.1 Distributed Sequential Pattern Mining

An abundant research has been done for finding sequential pattern mining in parallel and distributed areas like Hadoop, Grid, Cloud, etc. This section provides progress of algorithms for finding basic sequential patterns in distributed environment. The comparative study of these algorithms is done in section 4 on the basis of methodology used and evaluation parameters.

3.1.1 Parallel Transaction Decomposed Sequential Pattern Mining (PTDS)

In PTDS [30] transactions are decomposed to mine the sequential patterns and pattern growth approach is greatly accelerated to improve the efficiency of large scale data. First, PTDS sorts the sequences and plan the sequences with identical or similar prefix, which is considered as first transaction of each sequence. The input sequence is split in to two parts one is the first transaction and other is the remaining part of transaction in the sequence. PTDS collects sequences with equal prefix, decompose the prefix and applies serial sequential pattern mining method on the set of subsequences; each one contains the remaining transactions of the raw sequence, and finally merges the mining results together. PTDS is implemented using MapReduce framework on Apache Hadoop environment which greatly accelerate pattern growth approach and improves the performance and efficiency of parallel sequential pattern algorithm on large scale data.

3.1.2 CLAP: Collaborative pattern mining for distributed information system

Mining of data in distributed information system is divided into three parts are one identify locally important patterns on individual database, second determine major patterns after combining distributed database into single view and third find patterns which follow special relationship across different data collection. This algorithm [31] make use of pattern mining for query processing to satisfy user specified query constraints to discover patterns from distributed databases. In existing system pattern pruning is based on single database, so to solve this problem cross-database pruning concept is used for distributed sequential pattern mining. CLAP encourage pattern discovery in distributed approach where each distributed site carries pattern pruning in collaboration with its peers by employing bloom filter based pattern switching mechanism. A bloom filter is space efficient data structure which contains k hash functions, $H\_1(.), H\_2(.), \dots, H\_k(.)$ and binary array of m bits. Patterns like $x_1, x_2, \dots, x_n$ can be added into the bloom filter to check whether pattern exist in bloom filter or not by using all k hash functions to map x_t to k positions.

CLAP system consist of mainly two parts as one construction of FP-tree and bloom filter for each local site and second CLAP cross database pruning and pattern growth. CLAP only focuses on frequent itemset mining.

3.1.3 Mining Sequential Patterns on Grid computing environment

GSP algorithm which is apriori based approach is implemented on grid-computing environment. Apriori based algorithm are not having good performance but due to its loosely coupled processes, it is suitable to implement on distributed environment like grid. Two types of grids are designed data grid which is used to retain and provide data for mining while other is computing grid which is used to perform computing related job in sequential pattern mining. GSP is having five phases like first phase is sorting which sorts the database in order; second phase is litemset means collection of frequent items, third phase transformed the database by transforming frequent itemsets into unique

identification, fourth phase sequence of length where sequences are generated iteratively and fifth phase is maximal where frequent subsequences that are involved in other mined sequences are removed.

In grid environment all grids are installed with full functionality including all phases of GSP algorithm, each of which is wrapped by globus toolkit services. When user wants to perform sequential pattern mining, one computing grid is selected as master grid which takes care of all computing process of mining task and at least one data grid is select to retain the database. User can select one of the strategies for partitioning the mining task are as all mining tasks on single grid with complete or partial dataset and single mining task on single grid with complete or partial dataset. Through globus toolkit 3 wrapping, mining tasks are submitted to running client grids and message of progress on mining is returned and displayed on master grid. The algorithm is tested by changing the number of grids.

Mining sequential pattern in grid computing environment which uses divide-and conquer strategy and several monitoring mechanism for mining process are studied in [23].

3.1.4 Parallel Frequent Set mining using Inverted Matrix Approach

The new representation of transactional database is used in terms of Inverted matrix [32] which is distributed among the parallel nodes. For each parallel node the frequent item from the inverted matrix are assigned. Parallel nodes generates Co-Occurrence Frequent Item (COFI) tree for assigning frequent item. All nodes accomplish the mining process which generates all frequent items in which the assigned items are participated. Here the communication between the master node and all parallel nodes to generate all frequent itemsets is less.

There are two techniques used for assignment of frequent item to the parallel nodes, are one Alternate Loop Splitting (ALS), and second Block Loop Splitting (BLS). In ALS, all n frequent items from the inverted matrix where items are stored in ascending order of their support value are evenly assigned to m nodes, one by one

In BLS, for all n frequent items from the inverted matrix, first n/m items are assigned to node 1, next n/m items are assigned to node 2 and so on. It has been observed ALS achieves better time complexity as compared to BLS. Also both the parallel techniques are found to be better than sequential algorithm.

3.1.5 Sequential Pattern Mining on Cloud (SPAMC)

When dealing with big data traditional algorithm suffers from scalability problem. So to improve the scaling problem SPAMC [26] is developed for mining sequential patterns on MapReduce model on cloud. SPAMC is a cloud-based version of sequential pattern mining algorithm consisting of two phases: scanning phase, and mining phase.

In the scanning phase, high performance is achieved by distributing tasks on multiple computers by using MapReduce programming model to proceed in parallel by distributing sub-tasks to independent machines.

Each mapper scans and transforms a partitioned database, and reducers are used to count the frequency of each item and eliminate infrequent items. The bitmap information of frequent items will be stored into a distributed hash table (DHT) that can be accessed in the mining phase. After that, in the mining phase, the sequential pattern mining tasks are processed in parallel by distributed machines.

Main task of the mining phase is to construct the complete lexical sequence tree, and then all patterns can be derived. Additionally, to achieve better load balancing, depth first search strategy is used to bring out the steps of sequence and itemset extension with limited sub-tree depth.

This strategy effectively improves the situation like mapper may stand and wait for a long time. In such a context, each MapReduce round will complete two levels of lexical sequence sub tree construction. On the other side, reducers efficiently integrate output results from mappers and do the support counting to generate frequent sequential patterns of the current sub-tree.

3.2 Advanced Distributed Sequential Pattern mining

Based on unity of type advanced sequential pattern mining can be classified as closed sequence, maximal pattern, constraint based and time interval based sequential patterns. This section provides study of algorithms in advanced distributed sequential pattern mining.

3.2.1 Approximate Multidimensional Sequential Patterns on Distributed System

Approximate sequential pattern can be represented as, let $G_{x_1}, \ldots, G_{x_n}$ be the similar clusters for a local database DB_x then an approximate sequential pattern $lpat_{x_i}$ for group G_{x_i} is a sequence that minimizes the distance $dist(lpat_{x_i}, seq_a)$ for all seq_a in identical group G_{x_i}.

Multidimensional sequential pattern mining [33] extracts more useful information than sequential pattern mining as there are various applications use and access multidimensional database. In this paper, the multidimensional sequential patterns are converted to sequential patterns, by embedding the multidimensional information into the corresponding sequences. Then on distributed sites the sequences are grouped, summarized, analyzed, and the local frequent patterns are determined by the effective method of approximate sequential pattern mining. The multidimensional sequential patterns could be globally mined by high voting sequential pattern model after gathering all the local frequent patterns on one site. This multidimensional sequential pattern mining avoids mining of redundant information. After reducing the cost of communication, the global sequential patterns could be discovered effectively by the scalable method used in this paper. This approach solves the scalability problem of mining multidimensional sequential pattern but it brings the complexity.

3.2.2 Maximal Frequent itemsets from databases on Cluster of Workstations

Maximal frequent itemsets (MFI) is to find a large frequent itemset means long pattern early to avoid counting of all its subsets because all of them are frequent. All frequent itemsets can be obtained by finding all the maximal frequent itemsets.

This paper [34] proposed Distributed Mar-Miner (DMM) algorithm for mining of maximal frequent itemsets from databases. A frequent itemset is maximal if none its supersets is frequent. DMM requires very low synchronization and communication overhead in distributed computing systems.

DMM has mainly two phases are the local mining phase and the global mining phase. During the local mining phase, the local maximal frequent itemsets are discovered by mining local database on each node then they form a set of maximal candidate itemsets for the top down search in the subsequent global mining phase. A novel data structure prefix tree is developed to assist the storage and counting of the global candidate itemsets of different sizes. The global mining phase makes use of the prefix tree to work with any local mining algorithm. DMM is implemented on a cluster of workstations and evaluated its performance for various cases. DMM demonstrates better performance than other sequential and parallel algorithms, and its performance is also scalable for large maximal frequent itemsets in databases.

3.2.3 Time Interval based distributed sequential pattern mining

Sequential pattern mining with time intervals [35] [36] can be represented as Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of items. A transaction t is a list of alphabetically sorted items. A t has time-stamp information denoted by $t.time$. The time interval extended sequence t_s is a list of transactions with time interval and is defined as:

$t_s = \langle - D(t_1, t_1) - t_1, -D(t_1, t_2) - t_2, -D(t_1, t_3) - t3, \dots, -Dt1, tm - tm \quad t1.time \leq t2.time \leq t3.time \leq \dots \leq t_m.time.$

Here, $D\ (t_\alpha, t_\beta)$ is the time interval between t_α and t_β which is defined by expression:

$$D\left(t_\alpha, t_\beta\right) = d_n \mid n = \left\lfloor \frac{t_\beta.time - t_\alpha.time}{\Delta t} \right\rfloor$$

Δt is a user-defined parameter and determines the unit of time interval partition.

3.2.3.1 Profiling Node Conditions of Distributed System with Sequential Pattern Mining

Distributed monitoring systems normally generates huge amount of log data so that the problem of combining and summarizing the data is occurred. This paper [37] focuses on node conditions occurring in many computing nodes, which is called as node condition profile and can be considered as frequent occurring node condition patterns.

A sequential pattern mining with time interval is used for extracting computing node condition profiles from set of locally monitored results. Each node monitors its local node condition with m-monitor components and reports its monitoring result to aggregation node.

An aggregation node, executes sequential pattern mining for extracting node-condition profile, receives set of monitored results which are considered as one sequence from n-nodes. Extracted node condition profiles are useful to reduce the size of log data by replacing detailed logs to extracted profiles. As a result, the profiles help us to understand the condition of entire system easily.

3.2.3.2 Hiding prioritized sensitive patterns over distributed progressive sequential data streams

The key goal for privacy preserving distributed sequential data mining [38] is to permit computation of aggregate statistics over an entire data set without compromising the privacy of sensitive data of the participating or cooperative data sources.

On the basis of support counts and time at which the sensitive item set crosses the minimal threshold requirement hiding of sensitive patterns has been prioritized. There are different partitions of different distributed methods like horizontal, vertical and arbitrary are used to hide co-occurring sensitive patterns. Prioritized sensitive patterns are blocked into two phases.

In first phase, the source data at the collaborating party side hides the prioritized itemset which may disclose co-occurring sensitive patterns when all the itemsets of the blockset crosses the user threshold.

In second phase at trusted third party side; when we compute the integration of itemsets for collaborating parties, some of the newly produced co-occurring sensitive patterns may expose an important hidden knowledge were blocked. All the distributed partitions are effectively executed and tested with synthetic datasets.

3.3 Taxonomy of Distributed Sequential pattern mining

Figure 1 shows the taxonomy of distributed sequential pattern mining algorithms we have studied above.

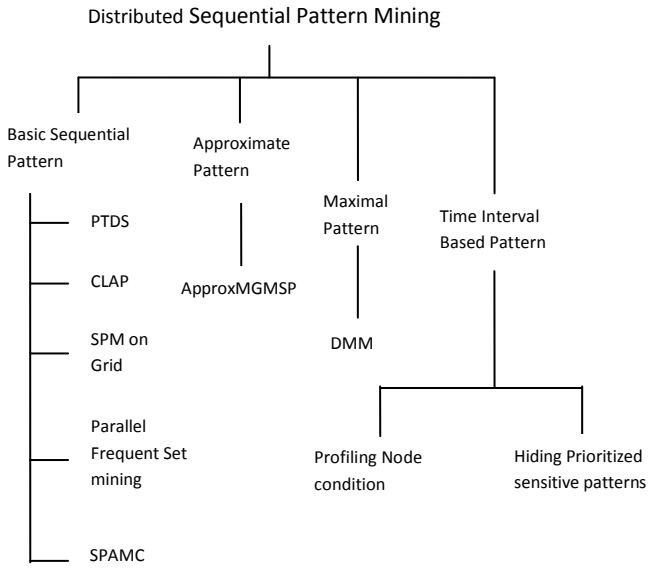

Figure 1: Taxonomy of Distributed Sequential Pattern Mining

4. COMPARATIVE STUDY

This section will provide the comparative study of basic distributed sequential pattern mining algorithms and advanced distributed sequential pattern mining algorithms.

4.1 Comparative study on Distributed sequential pattern mining

In this paper we already discussed different sequential pattern mining algorithms on distributed environment like PTDS, CLAP, SPM on Grid, Parallel frequent set mining and SPMAC.

Table 1 provides the comparative study of distributed sequential pattern mining algorithms. Comparison is based on environment used, method and various evaluation parameters used in respective algorithms. Experimental results of all the distributed sequential pattern mining algorithms with various approaches are discussed with advantages and limitations.

Table 1: Comparison of distributed sequential pattern mining algorithms

Algorithms	Parameters		
	Environment	**Methods**	**Evaluation Parameters**
PTDS	Apache Hadoop	Pattern Growth Approach and Transaction Decomposition	Execution Time, No. of computers
CLAP	Distributed Sites	Pattern Growth	Query Execution time
SPM on Grid	Grid	Apriori	Execution Time, No. of Grid Nodes
Parallel Frequent set Mining	Parallel Nodes	Inverted Matrix and COFI Tree	Execution Time, No. of Nodes
SPAMC	Cloud	Vertical Bitmap	Execution Time

PTDS and CLAP algorithms are based on pattern growth approach. The PTDS implementation and performance evaluation is done on Hadoop MapReduce framework. The sequence dataset considered is of China Mobile Corporation where number of sequences is 16 million, and the number of items is more than 200. The experimental results of the scalability of PTDS are compared with the PrefixSpan-based parallel method where it proves the better performance on different datasets since PTDS reduces the number of times of MapReduce passes and accelerates the pattern growth. Table 2 shows speedup and efficiency of PTDS algorithm

Table 2: Speedup and efficiency of PTDS algorithm

Number of computers	Time (second)	Speedup	Efficiency
32	1621	25.74	80.44%
64	899	46.33	72.40%
128	481	86.88	67.87%
256	305	138.9	54.30%

The experimental results of speedup are shown in Figure 2.The support threshold is set to 60%. The speedup of PTDS is increasing with the number of computers and is consistently higher.

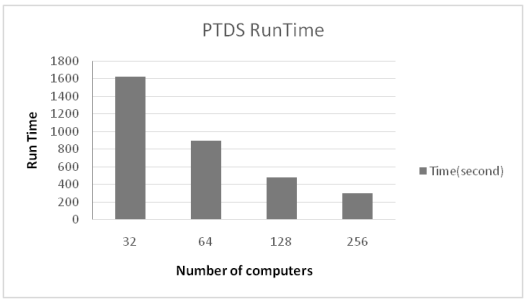

Figure 2: Speedup of PTDS

The performance of CLAP algorithm depends on depth limited pattern growth for cross-database pruning and bloom filter based message exchanging between sites. Experimental comparisons in the paper demonstrated that CLAP considerably outperforms other simple methods.
The implementation uses datasets of two groups (strong dense and weak sparse) of synthetic datasets generated from an IBM quest data generator [39] as shown in table 3. The system runtime mainly consists of constructing bloom filters containing length-l patterns for each site and the maximum collaborative mining time on a site. (l=3). Table 3 also shows overall system runtime.

Table 3: System runtime of CLAP algorithm

Database	CLAP(S_1) (Seconds)	System Runtime (Seconds)
Weak sparse	5.84	411.78
Strong Sparse	448.43	499.51

The CLAP distributed mining framework can be extended to handle other patterns, such as constrained frequent item-sets and closed frequent patterns.

In SPM on grid paper the test data sets are generated by the data generator to prove the effectiveness of GSP algorithm in grid computing environment. It uses globus toolkit as a grid computing environment. The experimental results show that the proposed grid computing environment provides a flexible and efficient platform for mining sequential patterns. In table 4 results show that the more grids for computing, the more speedups can be gained.

Table 4: Speedup of SPM on grid algorithm

Grid Node	Min. Support= 0.75		Min. Support= 0.3	
	Time	**Speed**	**Time**	**Speed**
16	1.08	1.17	481.96	484.59
8	1.01	1.05	462.24	464.73
4	1.02	1.07	425.86	428.15

Though the performance of SPM on grid is significant it can be further improved for data partitioning approach and job assignment strategies.

The parallel frequent set mining using inverted matrix and COFI tree is implemented using JAVA RMI. The parallel implantation is tested on a cluster of six nodes. Experiments are carried out on mushroom database.

As shown in Figure 3 and 4 experimental results are carried out for ALS and BLS algorithms. Figure 4 shows that ALS approach is efficient in terms of execution time than the BLS approach. Experimental results as shown in Figure 3 and 4 proved that, in both the techniques, as number of processing nodes increases, the processing time decreases, for the given support value.

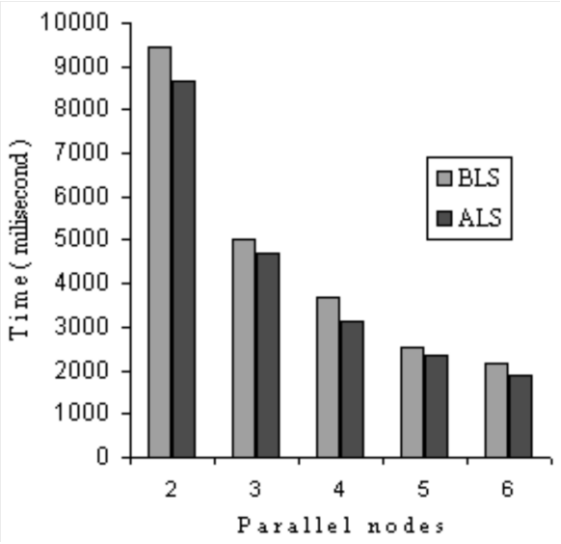

Figure 3: Execution time comparison of both parallel techniques for support=0.27

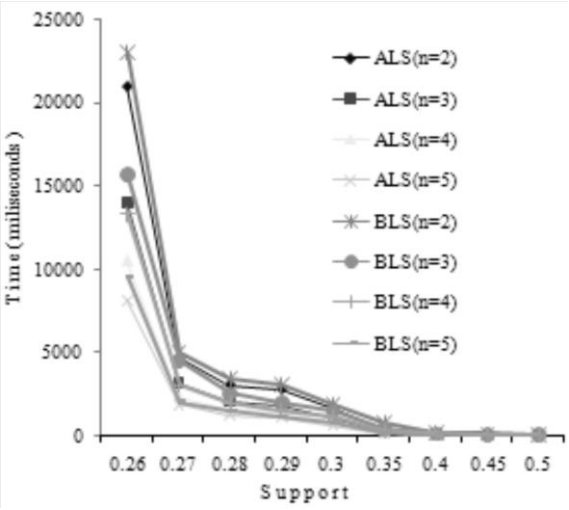

Figure 4: Average runtime vs. support

The SPAMC algorithm is implemented on Hadoop MapReduce framework in a cloud environment consisting of 32 machines. To verify the capability of sequential pattern mining on big data, experiments are performed to execute SPAMC and DPSP [40] on the synthetic dataset with up to 12.8 million transactions. DPSP is the state-of-the-art sequential pattern mining algorithm on the cloud computing environment.

Experimental results prove that SPAMC can significantly reduce mining time with big data, achieve extremely high scalability and load balancing in cluster environment.

4.2 Comparative study on advanced distributed sequential pattern mining types

Based on the unity of patterns classification of advanced distributed sequential pattern mining are approximate multidimensional, maximal patterns and time interval based patterns. The comparative study of sequential pattern types on distributed environment is done in this section. The comparison of all the reviewed algorithms is done on the basis of sequence type, environment used, methods and various evaluation parameters like execution time, dimensions, number of nodes.

In ApproxMGMSP an experimental result are carried out on synthetic dataset and it proves efficiency in mining multidimensional sequential patterns. For effectiveness analysis of ApproxMGMSP, a general evaluation method is applied that can evaluate the accuracy of the approximation in terms of how well it finds the real underlying patterns in the data and whether or not it generates any spurious patterns. The experiments show that algorithm is scalable but brings a high degree of complexity

In DMM it analyzes the characteristics of the algorithm in terms of speedup and sizeup. All tests were performed on synthetic datasets using 8-node cluster. The DMM largely reduces the number of synchronizations required between processing nodes. Experimental results show good speedup and sizeup for distributed mining of maximal patterns.

In Profiling Node Conditions of distributed system paper time interval based sequential pattern mining is proposed. The extracted profiles represent sets of event sequence with time interval which are frequently occurred at computing nodes. By using extracted node condition profiles, it is possible to not only reduce the size of log data by replacing detailed logs to extracted profiles. As a result, the profiles help us to understand the condition of entire system easily.

Table 5 shows comparative analysis of approximate, maximal and time interval based algorithmic approaches.

Table 5: Comparison of advanced distributed sequential pattern mining algorithms

Algorithms	Parameters			
	Sequence Type	**Environment**	**Methods**	**Evaluation Parameters**
ApproxMGMSP	Approximate	Distributed Sites	Multidimensional sequence mining	Execution Time, Dimensions
DMM	Maximal	Linux Cluster	Prefix Tree Structure	Execution time, Database size per node and No. of Nodes
Profiling Node Conditions	Time Interval	Distributed Nodes	Aggregation Mechanism	-
Hiding Prioritized Sensitive Patterns	Time Interval	Distributed Nodes	Horizontal, vertical and arbitrary data distribution	Length of Period of Interest

5. CONCLUSION AND FUTURE SCOPE

The intension of this paper is to review the distributed sequential pattern mining algorithms and advanced distributed sequential pattern mining algorithms. Our studies in sequential pattern mining on distributed area conclude that execution speed and scaling problem can be solved by distributed mining. The challenging issue to work with distributed environment is required scalable and powerful hardware to give better performance and finding the globally sequential patterns. Performance of distributed sequential pattern mining can be evaluated by increasing the size of database on each node, calculating the execution time and increasing the number of nodes.

There are various types of sequential pattern mining like approximate patterns, maximal pattern, constraint based, closed sequence, and time interval based patterns. More research is required in sequential pattern mining based on the advanced types like constraint based and closed sequences feature on distributed environment.

6. REFERENCES

[1] Agrawal R, Imielinski T, Swami A 1993. "Mining association rules between sets of items in large databases", in Proceedings of ACM-SIGMOD, Washington.

[2] R. Agrawal, R. Srikant, 1995. "Mining Sequential Patterns", in Proceedings of the International Conference on Data Engineering (ICDE), Taipei, Taiwan.

[3] Jiawei Han, Hong Cheng, Dong Xin, Xifeng Yan, 2007. "Frequent pattern mining: current status and future directions". Data Mining Knowledge Discovery.

[4] Han J., Dong G., Mortazavi-Asl B., Chen Q., Dayal U., Hsu M.-C., 2000. "Freespan: Frequent pattern-projected sequential pattern mining", in Proceeding of International Conference on Knowledge Discovery and Data Mining (KDD'2000).

[5] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, Mei-Chun, 2004. "Mining Sequential Patterns by Pattern-Growth : The PrefixSpan Approach". IEEE Transaction on Knowledge and Data Engineering.

[6] Liu Pei-yu, Gong Wei, Jia Xian, 2011. "An Improved PrefixSpan Algorithm Research for Sequential Pattern". In Proceedings of IT in Medicine and Education (ITME).

[7] Jinlin Chen, 2010. "An UpDown Directed Acyclic Graph Approach for Sequential Pattern Mining". IEEE Transaction on Knowledge and Data Engineering.

[8] Kamber M, Han J, Chiang JY, 1997. "Metarule-guided mining of multi-dimensional association rules using data cubes", in Proceedings of International conference on knowledge discovery and data mining, Newport Beach.

[9] Pasquier N, Bastide Y, Taouil R, Lakhal L, 1999. "Discovering frequent closed itemsets for association rules", in Proceedings of Seventh International conference on database theory (ICDT'99), Jerusalem, Israel.

[10] Pei J, Han J, Mao R, 2000. "CLOSET: an efficient algorithm for mining frequent closed itemsets', in Proceedings of ACM-SI0MOD international workshop data mining and knowledge discovery (DMKD'00), Dallas.

[11] Bayardo RJ, 1998. "Efficiently mining long patterns from databases", in Proceedings of ACM-SIGMOD international conference on management of data (SIGMOD'98), Seattle, WA.

[12] Grahne G, Lakshmanan L, Wang X, 2000. "Efficient mining of constrained correlated sets", in Proceedings of the International conference on data engineering (ICDE'00), San Diego, CA.

[13] Bonchi F, Lucchese C, 2004. "On closed constrained frequent pattern mining", in Proceedings of International conference on data mining (ICDM'04), Brighton, UK.

[14] Liu J, Paulsen S, Sun X, Wang W, Nobel A, Prins J, 2006. "Mining approximate frequent itemsets in the presence of noise: algorithm and analysis", in Proceedings of SIAM international conference on data mining (SDM'06), Bethesda, MD.

[15] Yan, X., Han, J., and Afshar, R., 2003. "CloSpan: Mining closed sequential patterns in large datasets". Third SIAM International Conference on Data Mining (SDM), San Fransico, CA.

[16] Chun-Sheng Wanga, Ying-Ho Liub, Kuo-Chung Chuc., "Closed inter-sequence pattern mining ". The Journal of Systems and Software, volume 86, 2013.

[17] Jian Pei, Jiawei Han, Wei Wang, "Constraint-based sequential pattern mining: the pattern growth methods", The Journal Intelligent Information System, Vol. 28, No.2, pp. 133 –160, 2007.

[18] Di Wu, Xiaoxue Wang, Ting Zuo, Tieli Sun, Fengqin Yang , 2010. "A Sequential Pattern Mining algorithm with time constraints based on vertical format". 2nd International Conference on Information Science and Engineering (ICISE), vol. no. 4, pp.3479-3482.

[19] Burdick D, Calimlim M, Gehrke J, 2001. "MAFIA: a maximal frequent itemset algorithm for transactional databases", in Proceedings of International conference on data engineering (ICDE'01), Heidelberg, Germany, pp 443–452.

[20] Luo C, Chung S. , 2005. "Efficient mining of maximal sequential patterns using multiple samples", in Proceedings of SIAM international conference on data mining (SDM'05), Newport Beach, CA, pp 415–426.

[21] Guralnik V, Karypis G., 2004. "Parallel tree-projection-based sequence mining algorithms". Parallel Computing, pp 443-472.

[22] Cong S, Han J, Padua D., 2005. "Parallel mining of closed sequential patterns", in Proceeding of Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, USA, pp. 562-567.

[23] Chih-Hung Wu, Yu-Chieh Lo, 2006. "Mining Sequential Patterns on a Grid-Computing Environment". IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan.

[24] Chih-Hung Wu, Chih-Chin Lai, Yu-Chieh Lo, 2012. "An empirical study on mining sequential patterns in a grid computing environment". Expert Systems with Applications, pp. 5748–5757.

[25] Shaochun Wu, Genfeng Wu, Shenjie Jin, 2004. "Pre-Clustering based Sequential Pattern Mining". Fourth

International Conference on Computer and Information Technology, pp. 1008-1013.

[26] Chun-Chieh Chen, Chi-Yao Tseng,Ming-Syan Chen, 2013. "Highly Scalable Sequential Pattern Mining Based on MapReduce Model on the Cloud". IEEE International Congress on Big Data.

[27] J. Ayres, J. Gehrke, T. Yu, and J. Flannick, 2002. "Sequential Pattern Mining Using a Bitmap Representation", in Proceedings of International Conference on knowledge Discovery and Data Mining, pp. 429-435.

[28] Wei Yong-qing, Liu Dong, Duan Lin-shan, 2012. "Distributed PrefixSpan Algorithm Based on MapReduce". International Symposium on Information Technology in Medicine and Education.

[29] Kong-Fa-Hu, Chang-Hai Zhang, Ling Chen, 2007. "A Scalable Method of Mining Approximate Multidimensional Sequential Patterns on Distributed Systems", in Proceedings of Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 19-22.

[30] Xueqiang Wang, Jing Wang, Tengjiao Wang, Hongyan Li, Dongqing Yang, 2010. "Parallel Sequential Pattern Mining by Transaction Decomposition". Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010).

[31] Xingquan Zhu, Bin Li, Xindong Wu, Dan He, Chengqi Zhang, 2011. "CLAP: Collaborative pattern mining for distributed information systems". Decision Support Systems 52, pp. 40-51.

[32] Sanjay D. Bhanderi, Sanjay Garg, 2012. "Parallel Frequent Set Mining Using Inverted Matrix Approach". Engineering (NUiCONE), Nirma University International conference.

[33] Changhai Zhang, Kongfa Hu, Zhuxi Chen, Ling Chen, Yisheng Dong, 2007. "ApproxMGMSP: A Scalable Method of Mining Approximate Multidimensional Sequential Patterns on Distributed System". Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007).

[34] Soon M. Chung, Congnan Luo, 2004. "Distributed Mining of Maximal Frequent Itemsets from Databases on a Cluster of Workstations". IEEE International Symposium on Cluster Computing and the Grid.

[35] Yu Hirate, Hayato Yamana, 2006. "Sequential Pattern Mining with Time Intervals". W.K.Ng, M. Kitsuregawa, J. Li(Eds) : PAKDD, LNAI, PP. 775-779.

[36] Fabian Morchen, 2007. "Unsupervised pattern mining from symbolic temporal data". ACM SIGKDD Explorations Newsletter, Volume 9 issue 1, Pages 41-55.

[37] Yu Hirate, Hayato Yamana, 2009. "Profiling Node Conditions of Distributed System with Sequential Pattern Mining". Software Technologies for Future Dependable Distributed Systems.

[38] Bettahally N. Keshavamurthy, Durga Toshniwal, Bhavani K. Eshwar, "Hiding co-occurring prioritized sensitive patterns over distributed progressive sequential data streams". Journal of Network and Computer Application, pp. 1116–1129,2007.

[39] IBM Quest Data Mining Project. Quest synthetic data generation code, http://www.cs.loyola.edu/~cgiannel/ assoc_gen.html.

[40] J. W. Huang, S. C. Lin, and M. S. Chen, 2010. "DPSP: Distributed Progressive Sequential Pattern Mining on the Cloud". Advances in Knowledge Discovery and Data Mining.

.