International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:3, 2014

A Distributed Approach to Extract High Utility Itemsets from XML Data

S. Kannimuthu, K. Premalatha

***Abstract*—**This paper investigates a new data mining capability that entails mining of High Utility Itemsets (HUI) in a distributed environment. Existing research in data mining deals with only presence or absence of an items and do not consider the semantic measures like weight or cost of the items. Thus, HUI mining algorithm has evolved. HUI mining is the one kind of utility mining concept, aims to identify itemsets whose utility satisfies a given threshold. Although, the approach of mining HUIs in a distributed environment and mining of the same from XML data have not explored yet. In this work, a novel approach is proposed to mine HUIs from the XML based data in a distributed environment. This work utilizes Service Oriented Computing (SOC) paradigm which provides Knowledge as a Service (KaaS). The interesting patterns are provided via the web services with the help of knowledge server to answer the queries of the consumers. The performance of the approach is evaluated on various databases using execution time and memory consumption.

***Keywords*—**Data mining, Knowledge as a Service, service oriented computing, utility mining.

I. Introduction

DATA mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods. Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction. With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps interpretation of such data and for the extraction of interesting knowledge that could help in decision-making. Data Mining, also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. While data mining and KDD are frequently treated as synonyms, data mining is actually part of the knowledge discovery process.

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Agrawal et al. [1] introduced association rules for discovering regularities between products in large scale transaction data recorded. The traditional ARM approaches [18], [19] consider the utility of the items by its presence in the transaction set. The frequency of itemset is not sufficient to reflect the actual utility of an itemset. For example, the sales manager may not be interested in frequent itemsets that do not generate significant profit.

The main objective of utility mining is to find all itemsets in a transaction database with utility values higher than the minUtilthreshold. Well known algorithms like Apriori [1] are available for mining association rules based on the support and confidence model. The frequency of an itemset may not be a sufficient indicator of interestingness, because it only reflects the number of transactions in the database that contain the itemset [2]. The utility can be measured in terms of cost, profit or other expressions of user preferences. For example, a computer system may be more profitable than a telephone in terms of profit.

Extensible Markup Language (XML) is emerging as a de facto standard for data exchanging in the internet due to XML's inherent data self-describing capability and flexibility of organizing data [4]. It's nested; self-describing structure provides a simple yet flexible means for applications to exchange data [3]. Most recently, XML is passing into virtually all areas of internet application programming, producing huge volumes of data encoded in XML. The ability to extract knowledge from XML data sources turned into a very important and necessary characteristic with the continuous growth in XML data. There have been increasing research efforts in mining frequent itemsets from XML data.

Data warehouse like Xyleme [5], [6] keeps all data in XML format which is integrated from the web. The integration of heterogeneous data sources in a traditional data warehouse faces challenges in physical level, syntactic level and semantic level. XML technologies and standard Application Programming Interfaces (APIs) are available to manage the heterogeneity conflicts that appear at the physical and the syntactic levels. XML technology like XSLT allows us to solve some semantic heterogeneity problems. The different attributes used to represent the same information at each local site can be translated to their common representation in the global schema with the help of XSLT. Many organizations adopt XML data wareshouse because of these benefits provided by the XML technologies. It is customary requirement to have direct methods for mining interesting patterns from XML data. So far, mining high utility itemsets requires mapping the XML data to the relational data model and using algorithms designed for relational databases to do the mining.

Data mining functionalities like Association Rule Mining

S. Kannimuthu is with the Coimbatore Institute of Engineering and Technology, Coimbatore, Tamilnadu, India (e-mail: kannimuthu.mer@gmail.com).

K. Premalatha is with the Bannari Amman Institute of Technology, Erode, Tamilnadu, India (e-mail: kpl_barath@yahoo.co.in).

(ARM), classification, clustering etc., in contemporary research involve mining of distributed, highly heterogeneous data. In the past years, standard data mining platforms, such as Weka [10], RapidMiner [11] and Orange [12], involve only their own data mining algorithms in the process of knowledge discovery from local data sources. Service-Oriented Computing (SOC) has emerged as a major research topic in the past few years. SOC is a new computing paradigm that makes use of services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications even in heterogeneous environments [13]. Although the concept has evolved from earlier component-based software frameworks, web service standards are based on the readily and openly available internet protocols, and thus are easier and cheaper for companies to adopt. A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It offers services as a platform-independent way that can be described, published, discovered, and loosely coupled in novel ways.

Next generation data mining technologies are focusing on enable processing of distributed data sources, the use of data mining algorithms implemented as web services, as well as the use of formal descriptions of data sources and knowledge discovery tools in the form of ontologies, enabling automated composition of complex knowledge discovery workflows for a given data mining task [43]. Standard data mining platforms such as Weka4WS and Orange4WS [43] are used in remote execution of the data mining algorithms through Web Services. In such a way, distributed data mining tasks can be concurrently executed on decentralized Grid nodes by exploiting data distribution and improving application performance. These tools are exposing the data mining algorithms as a Web service, which can be deployed on the available remote nodes.

The major contributions of the paper are mentioned as follows:

(1) The XML based implementation of HUI-Miner [8] algorithm is presented.
(2) The novel architecture for mining HUIs from the XML databases in a distributed environment is presented.
(3) The representation of weblog in a XML database and the way to extract the high utility itemsets in a XML database is given.
(4) Extensive experimental results are reported.

The remaining parts of the paper are organized as follows. In Section II, related works are mentioned. Section III discusses about problem definition. In Section IV, XML based implementation of HUI-Miner algorithm and the architecture of the proposed system are explained. Section V is concerned with the discussion of all the experimental results. Section VI concludes the work.

II. Related Works

The theoretical analysis of the utility mining problem presented by Yao et al. [14] lays the foundation for future utility mining algorithms. In their work, utility bound property and the support bound property are identified and mathematical model of utility mining is defined based on these properties. Chan et al. [15] presented a novel idea of top-K objective-directed data mining, which focuses on mining the top-K high utility closed patterns that directly support a given business objective. It is based on level-wise itemset mining approach which generates huge candidates which is filtered using a strategy based on utilities that allow pruning of low utility itemsets to be done by means of a weaker but anti-monotonic condition.

Liu et al. [16] propose the Two-Phase (TP) algorithm which efficiently extorts high utility itemsets from the databases. In Phase-I, transaction-weighted utilization is calculated and in Phase II, one database scan is performed to filter the high utility itemsets from high transaction-weighted utilization itemsets identified in Phase I. This algorithm restricts the size of the candidate set and simplifies the computation for calculating the utility. Liu et al. [17] suggested the parallel implementation of TP algorithm to mine high utility itemset from databases. Since TP algorithm is based on level-wise search approach which takes significant time in generating candidates. Liu et al. made use of Common Count Partitioned Database (CCPD) [18], [19] strategy to enable parallel implementation of two-phase algorithm.

Yao et al. [20] propose two algorithms UMining and UMining_H, which extract high utility itemset from the transaction databases. UMining uses level-wise approach to mine high utility itemsets. UMining_H finds most itemsets with utility values higher than minUtil based on a heuristic pruning strategy. This dissertation also proposes a pruning strategy called upper bound property to efficiently mine the high utility itemsets from the databases. A comprehensive study of utility based measures and a unified framework for utility based measures are given by Yao et al. The role and mathematical properties of utility based measures is discussed in [21].

A vital research issue extended from the utility mining is the discovery of temporal high utility patterns in data streams due to the wide applications on various domains. Temporal data mining can be defined as the activity of retrieving for interesting correlations or patterns in large sets of temporal data gathered for other purposes [22]. Vincent et al. [23] propose an algorithm named Temporal High Utility Itemset-Mine (THUI-Mine) that can discover temporal high utility itemsets from data streams efficiently and effectively. The fundamental idea of THUI-Mine algorithm is to incorporate the advantages of Two-Phase algorithm [24] and SWF algorithm [25] and augment with the incremental mining techniques for mining temporal high utility itemsets efficiently.

Hu and Mojsilovic [26] presented an algorithm for frequent item set mining that identifies high-utility item combinations. The main goal of the algorithm is to find segments of data, defined through combinations of few items (rules), which satisfy certain conditions as a group and maximize a predefined objective function. Hu and Mojsilovic devised the task as an optimization problem, presented an efficient

approximation to solve it through specialized partition trees, called High-Yield Partition Trees (HYP), and investigate the performance of different splitting strategies. The algorithm has been tested on "real-world" data sets, and achieved very good results. Earlier research on utility mining has been based on the candidate-generation and-test approach which is appropriate for sparse data sets with short pattern. Compressed Transaction Utility-Mine (CTU-Mine) proposed by the Erwin et al. [27] mines high utility itemsets using the pattern growth approach. CTU-Mine algorithm uses CTU-Tree data structure to efficiently mine high utility itemsets from dense and long patterns.

Most utility mining algorithms works well on sparse and short utility patterns but failed to extract the patterns efficiently in dense and long patterns. Erwin et al. [28] presented a new algorithm CTU-PRO that mines high utility itemsets by bottom up traversal of a Compressed Utility Pattern (CUP) tree. CTU-PRO algorithm has three major steps (1) Construction of LocalItemTable (2) Construction of Local CUP-Tree (3) Mining from Projection Database which are used to mine high utility itemset from the database efficiently. Weighted Association Rule Mining (WARM) considers the profits of items in a transaction database, such that the association rules about important items can be discovered. Nevertheless, high profit items may not always be high income products, since purchased quantities of items would also influence the revenue for the items. Yen and Lee [29] propose High Utility Quantitative Association Rules (HUQA) algorithm which considers both profits and purchased quantities of items to mine interesting patterns.

Apriori pruning strategy cannot be used to identify high utility itemsets. Li et al. [30] propose the Isolated Items Discarding Strategy (IIDS) to identify isolated items from transactions and ignore them in the process of candidate itemset generation. Two methods Fast Utility Mining (FUM) and Direct Candidates Generation+ (DCG+) were implemented to efficiently extract the high utility itemsets. Shankar et al. [31] proposed a novel algorithm called Fast Utility Mining (FUM) to mine high utility itemsets from the databases. FUM algorithm is a two step process: In first step, utility values for all single itemsets are calculated. In second step, generation of the candidates is done by scanning the transaction database once. The transaction which is already processed is ignored and finally high utility itemsets is retrieved by checking against minimum utility threshold.

Efficient data structures Incremental High Utility Pattern Lexicographic Tree (IHUPL), IHUP Transaction Frequency Tree (IHUPTF) and IHUP Transaction-Weighted Utilization Tree (IHUPTWU) recommended by Chowdhury et al. [32] to mine high utility itemsets in incremental databases. This article also elucidates the IHUP-Tree construction and algorithm to efficiently mine high utility itemsets. Effectiveness of the proposed data structures in different environment is experimented by comparing with existing algorithms such as TP [33], FUM [34], DCG+ [34].

Existing algorithms in utility mining failed to discover the exhibition periods of all items in a real world, for instance fresh products, seasonal products, and so on. Lan et al. [35] proposed a new kind of patterns, named On-shelf High Utility Itemsets (OHU), which consider not only individual profit and quantity of each item in a transaction but also common on-shelf periods of all items in a product combination in temporal databases. An approach called TP-OHUI (Two-Phase Algorithm for Mining On-shelf High Utility Itemsets in Temporal Databases) is also proposed by Lan et al. to efficiently discover on-shelf high utility itemsets. OHUI algorithm follows a new pruning strategy named on-shelf upper bound to filter early, the redundant candidate itemsets.

One of the challenges of utility mining is finding high utility itemsets with negative values from large databases. This issue has been investigated through Chun-Jung et al. [36] by proposing High Utility Itemsets with Negative Item Values-Mine (HUINIV-Mine) algorithm. HUINIV-Mine uses Two-Phase algorithm to mine high utility itemsets. Demanding research issue in data mining is to mine high utility patterns over data streams. Previous works [37], [38] of HUI mining were based on level-wise candidate generation and test problem. Hence, they require significant time and memory. Chowdhury et al. [39] proposed a new tree structure, called High Utility Stream tree (HUS-tree) and a innovative algorithm called HUP Mining over Stream data (HUPMS), for sliding window-based HUP mining over data streams. HUPMS algorithm uses a HUS-tree to mine all the high utility itemsets with a pattern growth approach.

Vincent et al. [40] suggested an algorithm Utility Pattern-Growth (UP-Growth) which makes use of new data structure UP-Tree. Four strategies Discarding Global Unpromising items (DGU), Discarding Global Node utilities (DGN), Discarding local unpromising items (DLU) and Decreasing Local Node utilities (DLN) introduced in this work to mine HUIs efficiently. Mining high utility itemsets from data streams is one of the most interesting research issues in stream data mining. Li [41] proposed an efficient algorithm called Mining High-Utility Itemsets based on LexTree-maxHTU (MHUI-max) employs effective representation of item information, called TID-list and a new lexicographical tree-based data structure (LexTree-maxHTU). Mining Web Access Sequences (WASs) can extract very useful knowledge from web logs with wide-ranging applications. Chowdhury et al. [42] presented a pioneering work to mine high utility WASs. Two tree structures, called utility-based WAS tree (UWAS-tree) and incremental UWAS-tree (IUWAS-tree) proposed for mining WASs in static and incremental databases.

III. Problem Definition

This work is best explained by using weblog database. In World Wide Web (WWW) and online services, if a user wishes to navigate from one object to another, they do via corresponding facilities (i.e., hyperlinks) provided. Understanding user access patterns from such an environment not only improves the system design but also be able to lead to better marketing decision (e.g. placement of advertisement in proper places, user classification and behavior analysis) [9]. The environment captures this type of user patterns are called

as mining traversal patterns. Conventional web usage pattern mining algorithms deal with only presence or absence of a web page and do not consider the semantic measures like weight or profit of the web page. Hence, high utility traversal pattern mining algorithm has evolved and it is based on HUI mining. So far, mining of utility based web access pattern from XML databases have not explored yet. This work addresses this issue. The web log data and utility data file is represented in XML format. The format of a sample web traversal XML transaction data file and XML utility data file is shown in Figs. 1 and 2 respectively. In transaction data file, the transactions tag is the root element that contains many weblog elements. Each weblog element is uniquely identified by its id attribute. Each weblog element contains one websites element which in turn contains many website elements. A website element has the name of the particular website in the given weblog and it has the time attribute. In utility data file, the websites tag is the root element that contains many website elements. Each website element is uniquely identified by its name attribute. Each item element has utility value.

```
<transactions>
<weblog id="T1"><websites>
<website time="1">A</website><website time="1">C</website><website time="1">D</website>
</websites></weblog>
<weblog id="T2">
<websites>
<website time="2">A</website><website time="6">C</website>
<website time="2">E</website><website time="5">G</website>
</websites></weblog>
<weblog id="T3"><websites>
<website time="1">A</website><website time="2">B</website><website time="1">C</website>
<website time="6">D</website><website time="1">E</website><website time="5">F</website>
</websites></weblog>
<weblog id="T4"><websites>
<website time="4">B</website><website time="3">C</website>
<website time="3">D</website><website time="1">E</website>
</websites></weblog>
<weblog id="T5"><websites>
<website time="2">B</website><website time="2">C</website>
<website time="1">E</website><website time="2">G</website>
</websites></weblog>
</transactions>
```

Fig. 1 Web traversal XML data file

```
<websites>
<website name="A">5</website>
<website name="B">2</website>
<website name="C">1</website>
<website name="D">2</website>
<website name="E">3</website>
<website name="F">1</website>
<website name="G">1</website>
</websites>
```

Fig. 2 External importance (utility) XML file

Let $I = \{I_1, I_2, I_3, \dots, I_n\}$ be a set of items and XDB be a XML database composed of transaction and utility information. $T = \{T_1, T_2, T_3, \dots, T_n\}$ be the set of transaction which is represented as a XML file (Refer Fig. 1). Each transaction $T_i \in T$ is a subset of$I$. The utility value of all items in I is represented as XML file (Refer Fig. 2).

$o(I_p, T_q)$ local transaction utility value, represents the quantity of item I_p in transaction T_q. For example, $o(A, T_1) = 1$, in Fig. 1. $s(I_p)$, external utility, is the value associated with item I_p in the Utility XML file. This value reflects the importance of an item, which is independent of transactions. For example, in Fig. 2, the external utility of item A, s(A) is 5.

$u(I_p, T_q)$, utility, the quantitative measure of utility I_p in transaction T_q, is defined as $o(I_p, T_q) \times s(I_p)$. For example, $u(A, T_1) = 1 \times 5$ in Fig. 1. $u(X, T_q)$, utility of an itemset X in transactionT_q , is defined as $\sum_{I_p \in X} u(I_p, T_q)$, where $X = \{i_1, i_2, \dots, i_k\}$ is a k-itemset, $X \subseteq T_q$ and $1 \leq k \leq m$. u(X), utility of an itemset X, is defined as

$$\sum_{T_q \in D \wedge X \subseteq T_q} u(X, T_q) \quad (1)$$

We find all the high utility itemsets using utility mining. An itemset X is a high utility itemset if $u(X) \geq \text{minUtil}$, where $X \subseteq I$ and minUtil is the minimum utility threshold. For example, in Fig. 1, $u(A, T_1) = 1 \times 5 = 5$, $u(\{A, C, D\}, T_1) = u(A, T_1) + u(C, T_1) + u(D, T_1) = 1 \times 5 + 1 \times 1 + 1 \times 2 = 8$, and $u(\{A, C, D\}) = u(\{A, C, D\}, T_1) + u(\{A, C, D\}, T_3) = 8 + 18 = 26$. If minUtil=40, then $\{A, C, D\}$ is not a high utility itemset. However, if an item is a low utility item, its superset may be a high utility itemset. Utility mining approach does not support downward-closure property [7]. Hence, we generate combinations of all items and same should be processed to ensure that no high utility itemset will be lost.

IV. Proposed Works

This section presents the proposed XML based implementation of HUI-MINER algorithm and the proposed architecture to mine HUIs in a service oriented computing environment.

Input:
-XML transaction data file
-External importance (utility) XML file
-minUtil threshold.

Output:
-High utility itemsets in a XML file

Step 1:Parse web traversal XML file using DOM parser

Step 2: Compute Transaction Utility (TU) for all the transactions.

Step 3: Calculate Transaction Weighted Utility (TWU) for all the items in the transactions.

Step 4: If the TWU of an item is less than a given minUtil, not consider the item for the subsequent mining process.

Step 5: Extract all the items whose TWUs exceed the minUtil and sort in ascending order in terms of TWU value.

Step 6: Construct the utility list by using [8]

Step 7: Extract high utility itemset by using HUI-Miner [8] algorithm.

Step 9: Return the HUIs in a XML format.

Fig. 3 HUI-MINER$_{XML}$ Algorithm

A. HUI-MINER$_{XML}$ Algorithm

The proposed HUI-Miner$_{XML}$ algorithm is best explained with an example. Consider the transaction database in Fig. 1 and the utility information in Fig. 2. Take *minUtil* threshold as 40. During the first scan of the database, TUs [33] of the transactions and TWUs [33] of the items are computed which is shown in Tables I and II, respectively. It is clearly understood in Table II that Items F and G are unpromising items since TWU of F and G are less than *minUtil* value (using TWDC property). These unpromising items are removed from the transaction T_2, T_3, T_5 as well as the utility of item F and G are also removed. The remaining promising items A, B, C, D and E in the transaction are sorted in the descending order of TWU. With this reorganized transaction database, utility list is constructed by using [8]. Finally HUIs are extracted from the reorganized XML transaction data file using HUI-Miner algorithm.

TWDC property: If TWU<*minUtil*, its supersets are unpromising to be high utility itemsets [33]

TABLE I
TU Value for All the Transactions in a Web Traversal XML Data File

TID	T_1	T_2	T_3	T_4	T_5
TU	8	27	30	20	11

TABLE II
TWU Value for All the Items in a Web Traversal XML Data File

Item	A	B	C	D	E	F	G
TWU	65	61	96	58	88	30	38

The main advantages of the proposed approach are

i) HUI-MINERXML algorithm extracts HUIs from XML databases. Thus heterogeneity problems in physical, syntactic and semantic level can be minimized.
ii) During the first scan itself unpromising items are removed by using TWDC property [33]. So Time taken to calculate the HUIs are significantly reduced.
iii) It consumes less amount of memory when compared to existing HUI-Miner approach (Since utility list does not have unpromising items).

B. Proposed Architecture

Fig. 4 exposes a SOC paradigm called Knowledge as a Service (KaaS), offers new types of service called HUI-MINER$_{XML}$, which is based on knowledge typically extracted from large volumes of data owned and maintained by different parties. Knowledge as a service provides data independency because the data are distributed among the different data owners, so it reduces the integration cost when compared to centralized- computing.

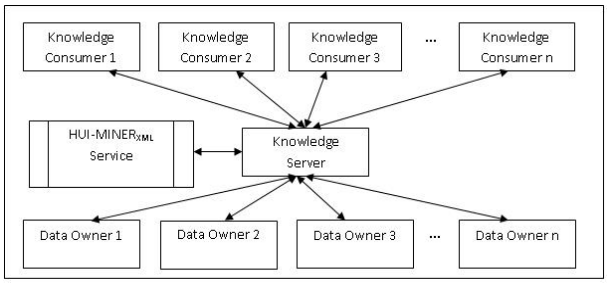

Fig. 4 Proposed Architecture of Knowledge as a Service (KaaS)

The proposed architecture (Fig. 4) visually depicts the working of the Knowledge as a Service in a distributed environment which makes use of HUI-MINER$_{XML}$ as a service. The subject of matter is how a consumer can get a data in a distributed environment where each and every data is distributed across several networks. Related data of some

particular domain applications may be distributed. The following steps have to be accomplished to access the distributed knowledge data which are being piled up in the repository of the data owners.

- The knowledge consumers have to communicate with the knowledge server systems requesting for the knowledge data without knowing the residing place of the knowledge data.
- Now, the knowledge server understands the user requests sent by the Knowledge consumers.
- After understanding the requests, the knowledge server will get to know which knowledge consumer has requested for the knowledge data and the server begins its search for the requested knowledge data from a pool of data owners available in the distributed environment using the web services.
- The Knowledge server can extract the knowledge data from the data owners with the help of Knowledge Extractor by implementing the HUI-MINER$_{XML}$ algorithm.
- The HUI-MINER$_{XML}$ algorithm extracts the knowledge data from the respective data owner at the backdrop and provides the knowledge data to the Knowledge extractor and then routes back the knowledge data to the knowledge server using web services.

This knowledge data is in-turn routed back to the corresponding knowledge consumer which requested for the knowledge data.

Advantages of the proposed architecture are:

i) **Transparency:** Knowledge extraction can be done without needing to understand detailed aspects of the underlying data mining algorithm. Furthermore, users can be able focus on the knowledge discovery application they must develop, without worrying about the SOA infrastructure and its low-level details.

ii) **Parallelism:** Architecture supports processing on large amounts of data through parallelism. HUI mining extraction are executed in parallel on different nodes, taking advantage at the same time of data distribution and web service distribution.

iii) **Interoperability:** This system uses web service technology. Interoperability is the main feature of web services in the case of SOAs.

iv) **Fault tolerance:** The system can continue to operation without interruption in the presence of partial network failures, or failures of the some software components, taking advantage of data distribution and web service distribution.

v) **Collaborative:** Utility mining tasks can be performed in collaborative mode with physically distributed participants.

V. Experimental Results

The experiments were conducted in the distributed environment which includes 5 slave sites and one master site. The transaction data are distributed equally for each slave site to compute HUIs locally. Finally these itemsets are consolidated by the master site to extract global HUIs. Nodes in the network have specification 2.40 GHz Intel® Core ™ i5-2430M CPU Processor with 4 GB RAM, and running on Windows 7. The algorithms were implemented in Java language. The data utilized in the experimental results are widely-accepted IBM synthetic data, Mushroom, Kosarak and Accidents which are converted into XML database due to the fact that there are no benchmark data available for this problem. The characteristics of these datasets are given in Table III. T10I4D10K in Table III denotes the Average size of the transactions (T), Average size of the maximal potentially large itemsets (I) and the number of transactions (D). Utility values for the items were assigned randomly in the utility XML data file. The execution time, number of HUIs and memory consumption is recorded for the IBM synthetic data, Mushroom, Kosarak and Accidents which are shown in Tables IV, V, VI, and VII respectively. The execution time of the HUI-Miner and HUI-MINER$_{XML}$ algorithm on all databases is depicted in Figs. 5-8. Execution time of the algorithm is recorded by varying minUtil threshold. It can be clearly shown that the proposed HUI-MINER$_{XML}$ algorithm takes less time than HUI-Miner algorithm. It is also observed that if the minUtil value is low, more number of the HUIs are generated and the execution time is high. For example, for database T10I4D10K in Fig. 5, when the minUtil thresholds are 5% and 10%, the numbers of high utility itemsets are 74631 and 14538, and the execution times of HUI-MINER$_{XM}$ are 113.30 seconds and 110.30 seconds, respectively.

TABLE III
Dataset Characteristics

Dataset	Size (KB)	No. of Transactions	No. of Items	AvgLen	MaxLen
T10I4D100K	6144	100000	1000	10.1	29
Mushroom	963	8124	119	23	23
Kosarak	49859	990002	41270	8.1	2498
Accidents	59663	340183	468	33.8	51

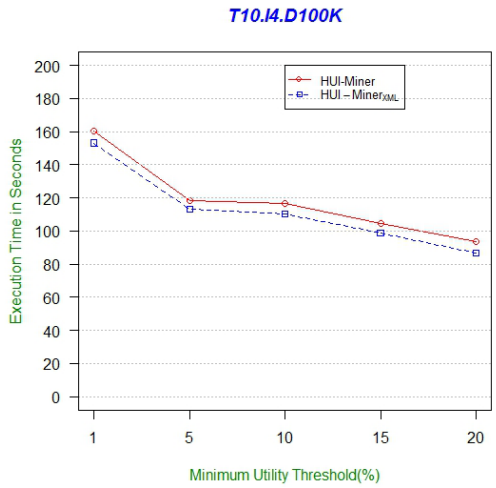

Fig. 5 Execution time on T10I4D100K

TABLE IV
RECORD OF NUMBER OF HUIS AND MEMORY CONSUMPTION BY VARYING MINUTIL THRESHOLD ON T10I4D100K

minUtil (%)	Number of HUIs generated	Memory consumption in MB	
		HUI-Miner	HUI-Miner$_{XML}$
1	74631	44.1	43.17
5	14538	40.83	39.29
10	5937	39.61	38.30
15	1344	37.6	36.99
20	339	32.14	31.76

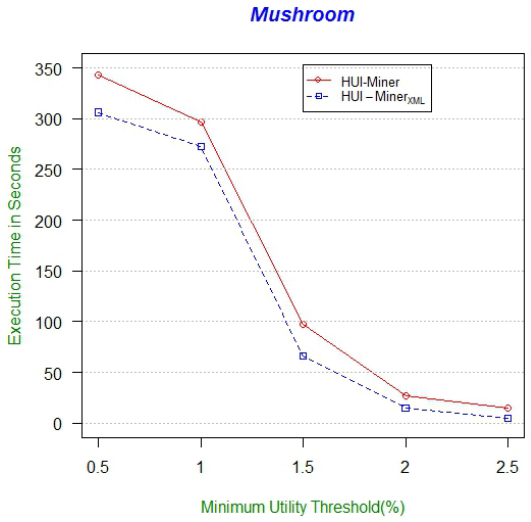

Fig. 6 Execution time on mushroom dataset

TABLE V
RECORD OF NUMBER OF HUIS AND MEMORY CONSUMPTION BY VARYING MINUTIL THRESHOLD ON MUSHROOM DATASET

minUtil (%)	Number of HUIs generated	Memory consumption in MB	
		HUI-Miner	HUI-Miner$_{XML}$
0.5	10023	29.31	28.46
1	8515	24.37	23.38
1.5	1540	22.81	21.35
2	828	22.9	21.02
2.5	7	21.26	20.15

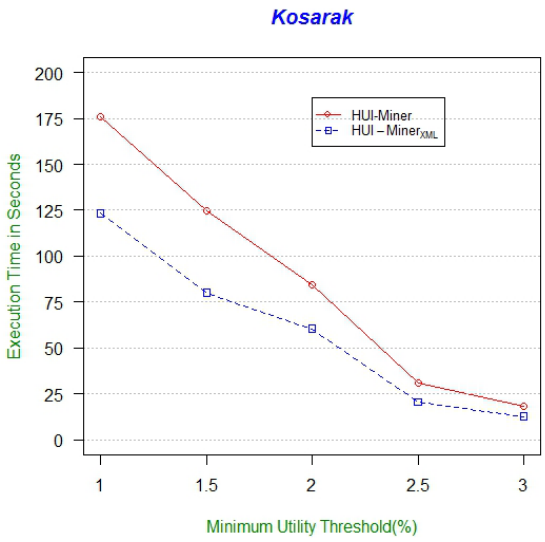

Fig. 7 Execution time on Kosarak dataset

TABLE VI
RECORD OF NUMBER OF HUIS AND MEMORY CONSUMPTION BY VARYING MINUTIL THRESHOLD ON KOSARAK DATASET

minUtil (%)	Number of HUIs Generated	Memory consumption in MB	
		HUI-Miner	HUI-Miner$_{XML}$
1	18673	101.14	100.40
1.5	5698	86.23	84.34
2	2275	67.02	66.32
2.5	1578	52.23	50.71
3	945	27.84	26.46

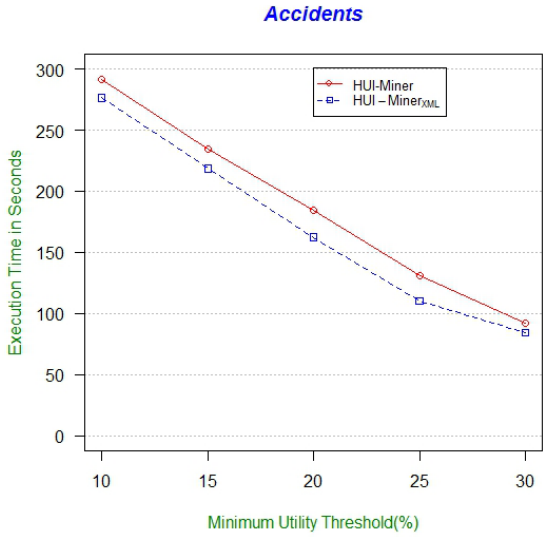

Fig. 8 Execution time on Accidents dataset

Existing algorithms in the utility mining literature are based on candidate generate-and-test approach and those algorithms consume a very large amount of memory to store candidate high utility itemsets during their mining processes. HUI-MINER$_{XML}$ approach extracts all HUIs without candidate generation and utilizes less amount of memory when

compared to the HUI-MINER algorithm. It is illustrated through Tables IV-VII.

TABLE VII
RECORD OF NUMBER OF HUIS AND MEMORY CONSUMPTION BY VARYING MINUTIL THRESHOLD ON ACCIDENTS DATASET

minUtil (%)	Number of HUIs Generated	Memory consumption in MB	
		HUI-Miner	HUI-Miner$_{XML}$
10	134687	343.71	340.12
15	94120	274.12	273.30
20	63104	212.05	210.56
25	30231	188.25	187.67
30	18733	114.9	112.50

VI. CONCLUSION

Extracting HUIs from the databases is an important task of data mining. It's a new and challenging area to perform mining on XML data due to the complexity of XML data. Existing algorithms work well on relational databases. In this work, the distributed approach for mining HUIs from XML databases is presented. The main intention of proposing this architecture is that the data integration cost is very much reduced because the KaaS provides better data independency as of when compared with the centralized environment. The paradigm of KaaS which we have proposed in this paper utilizes the HUI-MINER$_{XML}$ algorithm in distributed environment. The proposed approach was experimented by varying the minUtil threshold value for IBM synthetic dataset (T10I4D10K), Mushroom, Kosarakand Accidents and HUIs were obtained from the input XML database.

REFERENCES

[1] Agrawal, R., Imielinski, T., Swami, A.N., "Mining association rules between sets of items in large databases", In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, 1993, pp.207–216.
[2] Yao H and Hamilton J, Mining itemset utilities from transaction databases, Data &Knowledge Engineering 59, 2006, pp. 603-626.
[3] Uzair Ahmad, Mohammad Waseem Hassan, Arshad Ali, Richard McClatchey and Ian Willers, An Integrated Approach for Extraction of Objects from XML and Transformation to Heterogeneous Object Oriented Databases, Int. Conf. on Enterprise Information Systems (ICEIS) , 2003, pp. 445-449.
[4] W3C Consortium, http://www.w3.org, 2006
[5] Xyleme. http://www.xyleme.com.
[6] LucieXyleme. A dynamic warehouse for XML data of the web. IEEE Data Engineering Bulletin, 2001.
[7] Liu, Y., Liao, W.K., and Choudhary, A., "A two-phase algorithm for fast discovery of high utility itemsets", In Proc.of the Pacific Asia Conference on Knowledge Discovery andData Mining (PAKDD), 2005.
[8] Mengchi Liu and JunfengQu, "Mining High Utility Itemsets without Candidate Generation" In Proc. of the 21st ACM international conference on Information and knowledge management, 2012, pp. 55-64.
[9] Ming-Syan Chen, Jong Soo Park and Philip S. Yu, "Efficient Data Mining for Path Traversal Patterns", IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 2, 1998.
[10] http://www.cs.waikato.ac.nz/ml/weka/
[11] http://rapid-i.com/content/view/181/190/
[12] http://orange.biolab.si/
[13] DimitriosGeorgakopoulos and Michael P. Papazoglou, Service-Oriented Computing, The MIT PressCambridge, MassachusettsLondon, England, 2009.
[14] Hong Yao, Howard J. Hamilton, and Cory J. Butz.: "A Foundational Approach to Mining Itemset Utilities from Databases", In Proceedings of the 3rd SIAM International Conference on Data Mining, Orlando, Florida, 2004, pp. 482-486.
[15] Chan R, Yang Q and Shen Y.: "Mining high-utility itemsets". In Proc. of the 3rd IEEE International Conference on Data Mining (ICDM' 03). Melbourne, FL, 2003, pp. 19–26.
[16] Ying Liu, Wei-keng Liao, and AlokChoudhary: "A Fast High Utility Itemsets Mining Algorithm", UBDM'2005, 2005, pp. 90-99.
[17] Ying Liu, Wei-keng Liao, and AlokChoudhary: "A two-phase algorithm for fast discovery of high utility itemsets", In 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2005), Lecture Notes in Computer Science, vol. 3518, Springer-Verlag, Berlin, 2005, pp. 689–695.
[18] Zaki, M. J., Ogihara, M., Parthasarathy, S., and Li W. "Parallel Data Mining for Association Rules on Shared-memory Multi-processors. In Proceedings of the ACM/IEEE conference on Supercomputing, Pittsburg", PA, 1996.
[19] Zaki, M. J. "Parallel and Distributed Association Mining: A Survey. IEEE Concurrency, Special issue on Parallel Mechanisms for Data Mining", Vol. 7, No. 4, 1999, pp. 4-25.
[20] Hong Yao, Howard J. Hamilton.: "Mining itemset utilities from transaction databases, Data & KnowledgeEngineering", Elsevier Journal, Vol. 59, 2006, pp. 603-626.
[21] Hong Yao, Howard J. Hamilton, LiqiangGeng. : "A Unified Framework for Utility Based Measures for Mining Itemsets", Proceedings of the Second International Workshop on Utility-Based Data Mining, 2006, pp.28-37.
[22] Bruckhaus, T., Ling, C.X., Madhavji, N.H., and Sheng, S.: "Software Escalation Prediction with Data Mining". Workshop on Predictive Software Models, A STEP Software Technology & Engineering Practice, 2004.
[23] Vincent S. Tseng, Chun-Jung Chu, Tyne Liang. : "Efficient Mining of Temporal High Utility Itemsets from Data streams", In Proceedings of the Second International Workshop on Utility-Based Data Mining, 2006, pp. 18-27.
[24] Ying Liu, Wei-keng Liao, and AlokChoudhary: "A Fast High Utility Itemsets Mining Algorithm", UBDM'2005, 2005, pp. 90-99.
[25] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (editors). : "Advances in Knowledge Discovery and Data Mining", AAAI/MIT Press, 1996.
[26] Jianying Hu, Aleksandra Mojsilovic.: "High-utility pattern mining: A method for discovery of high-utility item sets", Elsevier Journal of Pattern recognition, Vol. 40, 2007, pp. 3317 – 3324.
[27] Alva Erwin, Raj P. Gopalan, N.R. Achuthan.: "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach", In Proceedings of 7thInternational Conference on Computer and Information Technology, 2007, pp. 71-76.
[28] Alva Erwin, Raj P. Gopalan, N.R. Achuthan.: "A Bottom-Up Projection Based Algorithm for Mining High Utility Itemsets", 2nd Workshop on Integrating AI and Data Mining (AIDM 2007), 2007, pp. 3-11.
[29] Show-Jane Yen and Yue-Shi Lee,: "Mining High Utility Quantitative Association Rules", DaWaK 2007, Lecture Notes on Computer Science, Springer-Verlag, 2007, pp. 283-292.
[30] Yu-Chiang Li, Jieh-Shan Yeh, Chin-Chen Chang, "Isolated items discarding strategy for discovering high utility itemsets", Data and Knowledge Engineering, Elsevier Journal, vol. 64, 2008, pp. 198-217.
[31] Shankar.S, Dr.Purusothaman.T, Jayanthi.S.: "Novel Algorithm for Mining High Utility Itemsets", In Proceedings of the 2008 International Conference on Computing, Communication and Networking (ICCCN 2008), 2008, pp. 1-6.
[32] Hong Yao, Howard J. Hamilton, LiqiangGeng. : "A Unified Framework for Utility Based Measures for Mining Itemsets", Proceedings of the Second International Workshop on Utility-Based Data Mining, 2006, pp.28-37.
[33] Ying Liu, Wei-keng Liao, and AlokChoudhary: "A two-phase algorithm for fast discovery of high utility itemsets", In 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2005), Lecture Notes in Computer Science, vol. 3518, Springer-Verlag, Berlin, 2005, pp. 689–695.
[34] Yu-Chiang Li, Jieh-Shan Yeh, Chin-Chen Chang, "Isolated items discarding strategy for discovering high utility itemsets", Data and Knowledge Engineering, Elsevier Journal, vol. 64, pp. 198-217.
[35] Guo-Cheng Lan, Tzung-Pei Hong, Vincent S. Tseng.: "Mining On-shelf High Utility Itemsets", International Conference on Information Technology and Applications in Outlying Islands, 2009, pp. 482-489.

[36] Jianying Hu, Aleksandra Mojsilovic.: "High-utility pattern mining: A method for discovery of high-utility item sets", Elsevier Journal of Pattern recognition, Vol. 40, 2007, pp. 3317 – 3324.
[37] Vincent S. Tseng, Chun-Jung Chu, Tyne Liang. : "Efficient Mining of Temporal High Utility Itemsets from Data streams", In Proceedings of the Second International Workshop on Utility-Based Data Mining, 2006, pp. 18-27.
[38] Chu, C.-J., Tseng, V.S., Liang, T.: "An Efficient mining for mining temporal high utility itemsets from data streams". Journal of Systems and Software, Vol. 81, 2008, pp. 1105–1117.
[39] ChowdhuryFarhan Ahmed, Syed KhairuzzamanTanbeer, Byeong-SooJeong, and Young-Koo Lee.: "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases, IEEE Transactions on Knowledge and Data Engineering", Vol. 21, no. 12, 2009, pp. 1708-1721.
[40] Vincent S. Tseng, Cheng-Wei Wu, Bai-En Shie and Philip S. Yu. : "UP-Growth: An Efficient Algorithm for High Utility Itemset Mining", In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, pp. 253-262.
[41] Hua-Fu Li.: "MHUI-max: An efficient algorithm for discovering high-utility itemsets from data streams", Journal of Information Science, vol.37, no. 5, 2011, pp. 532-545.
[42] Lee, K.L.,: "Efficient Graph-Based Algorithms for Discovering and Maintaining Knowledge in Large Databases", NTHU Master Thesis, 1997.
[43] Vid Podpecan, Monika Zemenova, Nada Lavrac, "Orange4WS Environment for Service-Oriented Data Mining", The Computer Journal 2011, doi: 10.1093/comjnl/bxr077.