

**MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY**

**BHOPAL (M.P.)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MINOR PROJECT**

**ON**

**PARALLEL AND WEIGHTED ITEMSET MINING BY MEANS OF MAPREDUCE FRAMEWORK**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF BACHELOR OF TECHNOLOGY

SUBMITTED BY: UNDER THE GUIDANCE

ABHIJEET RAWAT (141112110) OF:

CHUNDURU SRISAI GOUTHAM (141112101) DR**. NAMITA TIWARI**

KRISHNAPAL SINGH THAKUR (141112088) SESSION 2016-2017

LALU AHIRWAR (141112073)



**MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY**

**BHOPAL (M.P.)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that **ABHIJEET RAWAT**, **CHUNDURU SRISAI GOUTHAM**, **KRISHNAPAL SINGH THAKUR** and **LALU AHIRWAR** students of B.Tech 3rd Year (Computer Science & Engineering), have successfully completed their project “**PARALLEL AND WEIGHTED ITEMSET MINING BY MEANS OF MAPREDUCE FRAMEWORK**” in partial fulfilment of their minor project in Computer Science & Engineering.

**DR. NAMITA TIWARI**

(Project Guide)



**MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY**

**BHOPAL (M.P.)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DECLARATION**

We, hereby, declare that the following report which is being presented in the Minor Project Documentation entitled “**PARALLEL AND WEIGHTED ITEMSET MINING BY MEANS OF MAPREDUCE FRAMEWORK**” is the partial fulfilment of the requirements of the third year (sixth semester) Minor Project in the field of Computer Science And Engineering. It is an authentic documentation of our own original work carried out under the able guidance of Dr. Namita Tiwari. The work has been carried out entirely at Maulana Azad National Institute of Technology, Bhopal. The following project and its report, in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization.

We, hereby, declare that the facts mentioned above are true to the best of our knowledge. In case of any unlikely discrepancy that may possibly occur, we will be the ones to take responsibility.

**Name:**

ABHIJEET RAWAT (141112110)

CHUNDURU SRISAI GOUTHAM (141112101)

KRISHNAPAL SINGH THAKUR (141112088)

LALU AHIRWAR (141112073)

**ACKNOWLEDGEMENT**

With due respect, we express our deep sense of gratitude to our respected guide

Prof Namita Tiwari for her valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully. His rigorous evaluation and constructive criticism was of great assistance.

We are also grateful to our respected director Dr N. S. Chaudhary for permitting us to utilize all the necessary facilities of the college.

Needless to mention is the additional help and support extended by our respected HOD, Dr R. K. Pateriya, in allowing us to use the departmental laboratories and other services.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind co-operation and help.

Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing the much needed support and encouragement.

**CONTENTS**

Declaration

Acknowledgement

Abstract……………………………………………………………………

1. Introduction…………………………………………………………………………………………….

2. Big Data Concepts…………………………………………………

BigData ……………………………………………………

Hadoop ……………………………………………………

MapReduce ……………………………………………………

4. Parallel weighted itemset mining from BigData ………………………………………

Data preparation………………………………………………

Weighted Itemset Mining…………………………………

5. Implementation……………………………………………………………………………………...

Algorithms ………………………………………………………

User Interface …………………………………………………

6. Hardware and Software Requirements

Conclusions and Future work

References

**Abstract**

Frequent itemset mining is an exploratory data mining technique that has fruitfully been exploited to extract recurrent co-occurrences between data items. Since in many application contexts items are enriched with weights denoting their relative importance in the analysed data, pushing item weights into the itemset mining process, i.e., mining weighted itemset rather than traditional itemset, is an appealing re-search direction. Although many efficient in-memory weighted itemset mining algorithms are available in literature, there is a lack of parallel and distributed solutions which are able to scale towards Big Weighted Data. This project presents a scalable frequent weighted itemset mining algorithm based on the MapReduce paradigm. Weights indicate the ratings given by users to the purchased items. The mined itemset represent combinations of items that were frequently bought together with an overall rating above average.

**Introduction**

Data storage has increased exponentially in the world over the past few years. Data coming from different sources such as web logs, machine logs, human generated data etc. are being stored by companies. This phenomenon is known as "Big Data" and now a days is trending everywhere. With the incredibly fast growth of data comes the need to analyse the huge amount of data. Due to lack of adequate tools and programs, data remains unused and underutilised with many important knowledge useful for mankind getting hidden. For processing huge amount of data, new tools and approaches have evolved over period of time. Generally, the most commonly used tool for analysis of Big Data is Hadoop. It is a very handy tool for batch processing of large datasets. Mining of frequent patterns of itemsets found in large transactional datasets helps in further important data mining tasks like association rule mining, correlations, clustering etc.

In real-life applications items are unlikely to be equally important within the analysed data. For example, items purchased at the market have different prices, medical treatments have different urgency levels, and genes are expressed in biological samples with different levels of significance. Hence, an appealing extension of traditional itemset mining algorithms is to push of item relevance weights into the mining process. We propose Parallel Weighted Itemset miner, a new parallel and distributed framework to extract frequent weighted itemsets from potentially very large transactional datasets enriched with item weights. The framework relies on a parallel and distributed-based implementation running on a Hadoop cluster. To make the mining process scalable towards Big Data, most analytical steps performed by the system are mapped to the MapReduce programming paradigm.

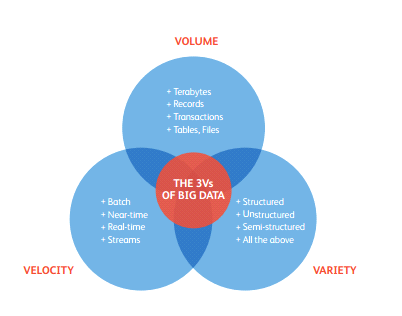
**BigData concepts**

**About BigData**

Big Data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyse. This definition is intentionally subjective and incorporates a moving definition of how big a dataset needs to be in order to be considered big data.

This definition emphasizes that there is no concrete volume of data to be considered as ‘big’, but it depends on the context. Here the definition uses size or volume of data as the only criterion. The term ‘Big Data’ can be misleading at times as it suggests that the notion is mainly about the volume of data. But considering the interest that big data bring in there is obviously more about it rather than just volume of data. Therefore another interesting definition of Big Data, is from IDC: Big Data technologies as a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data by enabling high-velocity capture, discovery, and/or analysis. There are three main characteristics of Big Data: the data itself, the analytics of the data, and the presentation of the results of the analytics.

This definition is based on the 3Vs model coined by Doug Laney in 2001. He did not use the term Big Data but predicted that data management will get more and more important and difficult. He then identified the 3 Vs - data volume, data velocity and data variety as the biggest challenges of data management.



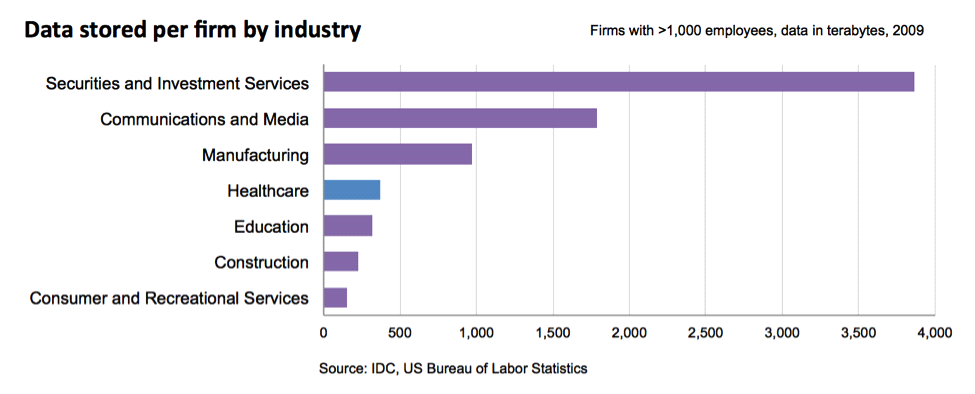
Data volume is the size of the data, data velocity is the speed at which data arrives and data variety is the data extracted from different sources which can be unstructured or semi-structured. Some researchers slightly modify the 3Vs model. Sam Madden describes Big Data as data that is ‘too big, too fast and too hard’, where ‘too hard’ refers to data that does not fit into existing data processing tools. Kaisler defines ‘Big Data’ as the amount of data just beyond technology’s capability to store, manage and process efficiently, but mention variety and velocity as additional characteristics. Tim Kraska moves away from the 3Vs model, but still acknowledges the fact that ’Big Data’ is more than just volume. He describes ‘Big Data’ as data for which ‘the normal application of current technology doesn’t enable users to obtain timely, cost-effective, and quality answers to data driven questions’. The 3Vs model is overall the most widely used and accepted description of what ‘Big Data’ means.

**How Big is Data**

In this section, we try to give an idea about the rapid growth of data in the past few years. As technology advances over time, we assume the size of datasets that qualify as Big Data will also increase. Today data in many sectors will range from a few dozen terabytes to multiple petabytes.

**Globally**

Two researchers Hal Varian and Peter Lyman from University of California Berkeley as part of their project "How much Information?" estimated that about 5 Exabyte (100000 Terabytes) of data were globally stored in 2002 and about three times of this amount, 18 Exabyte were transmitted not necessarily stored through electronic channels as radio, telephone, television and internet. They also found out that the amount of data stored doubled during the period of 1999 to 2002, a compound annual growth of about 25 percent. In 2007, an annual series of reports by a research firm IDC on the "Digital Universe" about the amount of data created and replicated each year revealed that the amount of digital data created in a year exceeded the world’s data storage capacity. The rate of data generation is increasing much faster than the expansion of world’s data storage capacity. The study of IDC in 2009 estimated that the total amount of data created and replicated was 800 Exabyte and surpasses 1.8 zettabytes in 2011 growing by a factor of 9 in just five years. They also projected that the volume of data would grow by more than 40 times by 2020, an annual growth rate of about 40 percent. Figure represents this global explosion of data through a graph.



**By Sectors**

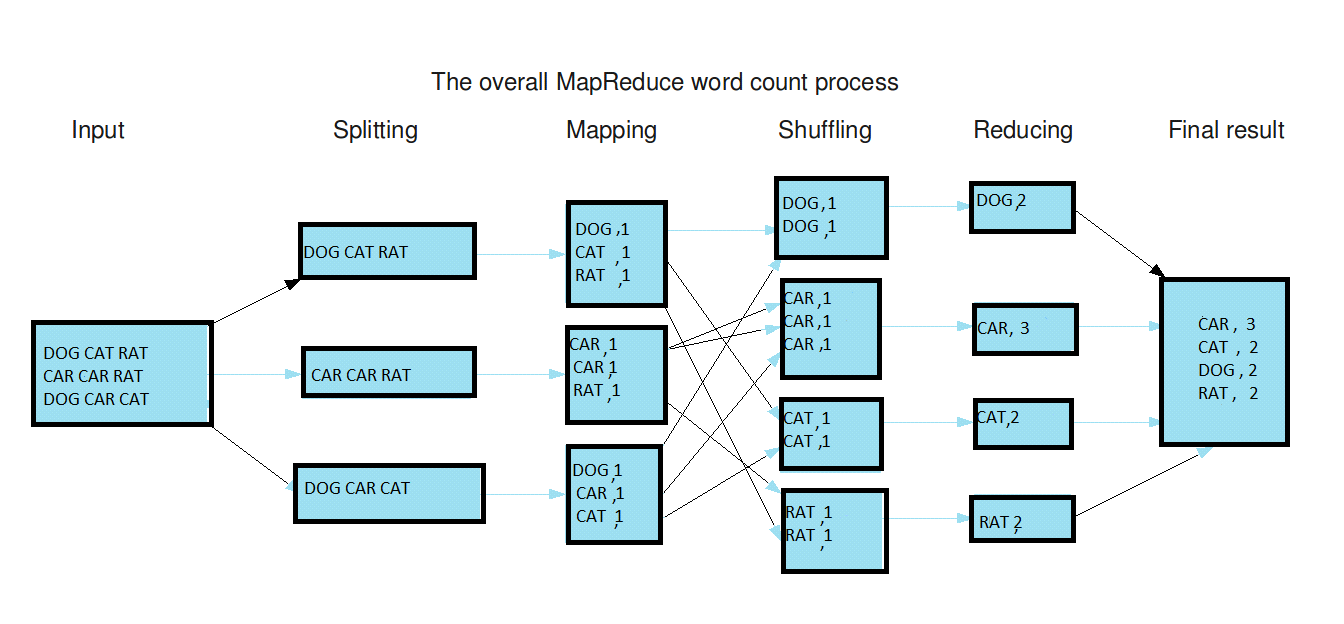
The growth of Big Data is a common trend observed in every sector. There is a global acceptance that data generation is growing exponentially. McKinsey Global Institute estimated that, by 2009, almost all sectors in the US economy had an average of 200 terabytes of stored data per company with more than 1000 employees. In figure 2.3 financial services sectors, including investment services, securities and banking, have on average the most digital data per firm (for example, New York Stock Exchange boasts about half a trillion trades per month). Government sectors, communications, media firms and utilities also have a significant amount of data stored per enterprise or organization. Discrete and process manufacturing have the highest aggregate amount of data stored.

**Hadoop Framework**

Hadoop framework is allows data storing and running applications on clusters of commodity hardware. It provides massive storage for any kind of data. Hadoop is the parallel programming platform built on Hadoop DISTRIBUTED File Systems (HDFS) for MapReduce computation. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS holds very large volume of data and provide easier access. HDFS also makes applications available to parallel processing. HDFS is a part of Apache Hadoop main project.

**MapReduce**

MapReduce is a programming model and an associated implementation for processing and generating large datasets. Users specify a map and reduce functions, they takes <key, value>pair as an input and generates intermediate <key, value>pairs and merges all intermediate values associated with the same intermediate key respectively.



Weighted Itemset Mining System Architecture This paper presents a scalable frequent weighted itemset mining algorithm based on the MapReduce paradigm. To demonstrate its action ability and scalability, the proposed algorithm will be tested on a Big dataset collecting large volume of reviews of items. Weights indicate the ratings given by users to the purchased items. The mined itemsets represent combinations of items that were frequently bought together with an overall rating above average. It integrates a variant of the BigFIM algorithm which is able to successfully cope with data enriched with weights. Furthermore, to allow experts to effectively explore the result of the mining process, the proposed system allows us to rank itemsets by (i) weighted support, (ii) traditional support, and (iii) a mix of the above. While the traditional support indicates the generic degree of interest of the considered combination of items, the weighted support integrated in the proposed framework indicates the average level of interest of the least interesting item within each transaction. The proposed system, running on a Hadoop cluster, overcomes the limitations of state-of-the art approaches in coping with datasets enriched with item weights.

**Modules Split Up**

Module 1: Registration module

This module is used for the user to register their login id by providing the minimal information. So that they can login to the website.

Module 2: Sign in module

In this, user can login to the website by registered login id and a valid password. Only the authenticated user can login and use the website.

Module 3: Data server module

In this module, user extracts itemset which are frequently brought together with an overall rating above average.

**Parallel Weighted Itemset Mining**

**From BigData**

Parallel Weighted Itemset Miner is a new data mining environment aimed to analyse Big Data equipped with item weights. The main environment blocks are briefly introduced below. A more detailed description is given in the following section.

**Data preparation**

This step entails preparing data to the subsequent item set mining process. The source data is acquired, stored in a transactional dataset, and equipped with item weights. A transactional dataset is a set of transactions. Each transaction is a set of (not repeated) terms. Depending on the context of analysis, items may represent different concepts (e.g., products, objects, places, stocks). For example, let us consider the dataset reported in Table I. It is an example of (unweighted) transactional dataset consisting of five transactions, each one representing a different customer of an e-commerce company. For each customer the list of purchased items is known. For instance, customer with id 1 bought items X, Y, and Z. Note that each transaction, which represents a distinct electronic basket, may contain an arbitrary number of items. To consider the relative importance of the items within each transaction during the item set mining process, itemset enriched with weights. A transactional dataset whose items are equipped with weights is denoted as weighted transactional dataset.

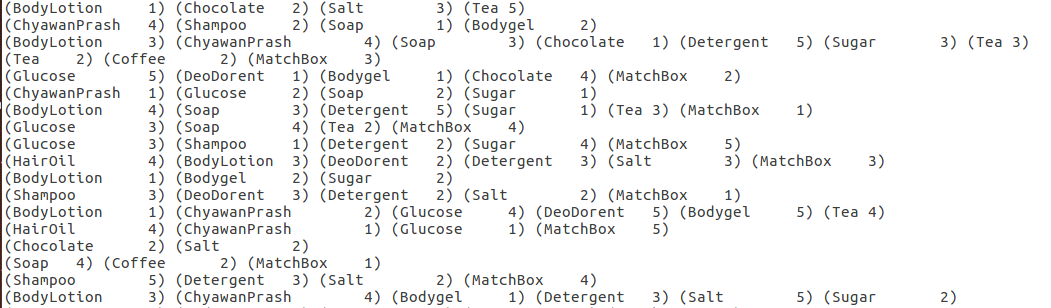
TABLE I: EXAMPLE OF UNWEIGHTED DATASET: ITEM BOUGHT BY CUSTOMERS

|  |  |
| --- | --- |
| **Customer ID** | **Purchased item** |
| 1  2  3  4 | X,Y,Z  X,Y,Q  X,Y,Z  X,Z,Q |

TABLE II: EXAMPLE OF WEIGHTED DATASET: ITEM RATINGS GIVEN BY CUSTOMERS

|  |  |
| --- | --- |
| **Customer ID** | **Purchased items and ratings** |
| 1  2  3  4 | (X, 3) (Y, 1) (Z, 5)  (X, 2) (Y, 2) (Q, 2)  (X, 4) (Y, 2) (Z, 5)  (X, 3) (Y, 3) (Q, 2) |

Dataset used by PaWI Framework



Item is a pair \_item, weight\_, where weight is the weight associated with the corresponding item. For example, let us consider the weighted transactional dataset reported in Table II. It extends the traditional transactional dataset in Table I by enriching items with the corresponding weights. More specifically, for each customer the rating (from one to five) given to each purchased item is known. For instance customer with id 1 rated item X as 3, item Y as 1, and item Z as 5. The analysed data are tailored to a weighted transactional data format. Furthermore, if need be, ad hoc pre-processing steps are applied to the raw data to ensure high-quality results. Data filtering entails discarding the items/transactions that are irrelevant for subsequent analyses. For instance, recalling the previous example, duplicate entries of the same customer basket can be removed because they could bias item set support counts. To ensure the scalability of the knowledge discovery process, the PaWI system performs data filtering as a distributed MapReduce job.

**Weighted Itemset Mining**

This step focuses on mining frequent weighted itemsets from the prepared weighted dataset. A k- itemset (i.e., an itemset of length k) is a set of k items. The traditional support value of an itemset in a transactional dataset is given by its frequency of occurrence in the source dataset. For example, {X, Y} is an itemset indicating the co-occurrence of items X and Y. If we disregard item weights, this itemset has a support equal to 4 in Table I because it occurs in four out of five transactions, meaning that most of the users purchased items X and Y together. The goal of this paper is to extend traditional large-scale itemset miners to successfully cope with big weighted data. Hence, for our purposes, the itemset support measure is extended, similar to, to the case of weighted data. The weighted support of an itemset 1 in a weighted transactional dataset T is defined as a linear combination of the aggregation weights computed on each transaction in T. An arbitrary aggregation function f (e.g., min, max, average, and mode) can be potentially applied to aggregate item weights within each transaction. The choice of f depends on the considered use cases. Hereafter, we will consider f=min (i.e., the least weight of any item in I is considered), because, as discussed, the selected patterns are deemed as particularly useful for analysing real Big datasets. Recalling the running example, let us consider analysts who would like to discover the combinations of items that were frequently bought together with an overall rating above average. To this aim, we may consider item ratings during support computation by weighting item set occurrences within each transaction by the least item rating. For instance, recalling the weighted transactional dataset in table for customer with id 1 between X and Y the item with least rating is Y (rating equal to 1), while for customer with id 5 is X (rating equal to 2). Hereafter we will denote as weighted support the support of an item set by considering item weights, whereas as traditional support the item set support disregarding item weights. For instance, {X, Y} has weighted support equal to 1+2+2+3+0=8 and traditional support equal to 4. Given a weighted transactional dataset D and an (analyst provided) minimum support threshold minus, the PaWI system addresses the extraction of all frequent weighted item sets from D. To allow comparing weighted item sets with traditional ones, PaWI allows experts to mine traditional item sets as well. As discussed below, the support thresholds enforced during weighted and unweighted item set mining are potentially different. The weighted item set mining process relies on a parallel and distributed-based algorithm running on a Hadoop cluster. To make the mining process scalable towards Big Data, the mining steps are mapped to the MapReduce programming paradigm. MapReduce is a parallel programming framework providing both a relatively simple programming interface together and a robust computational architecture. MapReduce programs consist of two main steps. In the map step, each mapper processes a distinct chunk of the data and produces key- value pairs. In the reduce step, key-value pairs from different mappers are combined by the framework and fed to reducers as pairs of key and value lists. Reducers generate the final output by processing the key/value lists. To efficiently perform frequent weighted item set mining with MapReduce PaWI integrates a variant of the BigFIM algorithm which is able to successfully cope with data enriched with weights. The exploitation of weights is challenging because ad-hoc data structures must be used to efficiently maintain the weights associated with each item and transaction by balancing the impacts on computational and communication costs. The following extensions have been proposed: Distributed transaction splitting. BigFIM relies on two established item set miners: Apriori and Eclat. We modified the BigFIM algorithm to allow both Apriori and Eclat to successfully cope with weighted data. More specifically, our algorithm generates an equivalent version of the source dataset that includes only transactions with equally weighted items. Let us assume that the weight of an equivalent transaction tq is w. Then, the occurrence of any item set in tq will be weighted by w instead of by 1. Each transaction in the original dataset may correspond to a set of equivalent transactions in the equivalent dataset. A formal definition of the equivalence set of weighted transactions is given in Note that since two distinct transactions have disjoint equivalent sets the splitting process is straightforwardly parallelizable. Weighted support counting. Since items are equipped with weights, traditional support counting is replaced with weighted support counting, according to Definition 1. To accomplish itemset support counting different strategies are adopted according to the algorithm used. Specifically, to perform Apriori-based weighted itemset mining, item set supports are counted by generalizing the word counting problem for documents to weighted item sets, i.e., each mapper receives a disjoint subset of dataset transactions (i.e., the documents) and reports the items/item sets (i.e., the words) for which the weighted support count is performed. A reducer combines all partial weighted support counts and reports only the items/item sets whose weighted support is above the threshold. These frequent weighted item sets are redistributed to all mappers to act as candidates for the next step of breadth-first search and then the procedure is repeated to mine weighted item sets of higher length. To perform Eclat-based weighted itemset mining, each mapper builds the weighted tidlist of the item sets related to a subset of transactions. The weighted tidlist of an arbitrary item i j consists of all pairs (transaction id, weight) such that the transaction related to transaction id contains item i j with weight. For example, let us assume that a mapper receives the transactions contained in the dataset in Table II. For item Z it generates the following weighted tidlist: {(cid1,5),(cid3,5),(cid5,4)}. The weighted tidlist consists of all pairs (customer id, weight) for which the transaction related to customer id contain item Z with weight. For instance, the transaction corresponding the electronic basket of the customer with id 1 contains item Z with weight 5. A reducer combines all partial weighted support counts and reports only the items/item sets whose weighted support is above the threshold. Note that the equivalent transaction weights are not stored in the distributed cache, because BigData sets may potentially consist of millions of transactions.

**Item set Ranking**

The manual exploration of all the item sets (weighted or not) mined from Big data is practically unfeasible. Hence, to support the knowledge discovery process experts may would like to access only a subset of most interesting patterns. This step focuses on ranking the mined item sets according to their level of significance in the analysed data. To filter and rank the mined item sets, the support measure is the most commonly used quality index. To cope with weighted data, for each candidate itemset the PaWI system computes both the traditional and weighted support measures. While the traditional support value indicates the observed frequency of occurrence of the considered combination of items in the source dataset, in weighted support counting itemset occurrences are weighted by the least item weight (see Definition 1). To select item sets whose average least item weight is maximal the PaWI system combines the weighted and traditional support measure in a new measure called AW sup, i.e., the Average Weighted support. The AW-sup measure is defined as the ratio between the weighted itemset support and the traditional itemset support. It indicates the average per-transaction weight of the least weighted item. Selecting top interesting item sets based on this measure is potentially interesting in real applications. For example, let us consider again the example dataset in Table II. According to Definition 1, itemset{X,Y} has weighted and traditional support values equal to 8 and 4, respectively. Since transactions represent electronic baskets, the weighted itemset support indicates the overall least item rating computed on the subset of customers who bought both items X and Y , while the traditional support measure indicates the simple frequency of occurrence of the combinations of items in the electronic basket dataset. The AW-sup value of {X,Y} is 2, meaning that, on average, for each electronic basket containing items X and Y both items have been rated at least 2. Ranking the mined item sets by decreasing AW-sup allows experts to consider first the combinations of items that got maximal average ratings. Note that this statistics cannot be straightforwardly computed based on simple averages, because (I) it considers only the electronic baskets containing both X and Y , (ii) for each basket it selects the rating of the least weighted item between X and Y . Item sets not satisfying the traditional support threshold are discarded because they represent combinations of items that rarely occur in the analysed data. The setting of the minimum weighted support threshold is driven by the average rating of the selected items. More specifically, we are interested in exploring the frequent combinations of items with rating above average, i.e., the item sets whose AW-sup is above a minimum threshold.

**Implementation**

The below steps gives an overview about weighted item set mining.

Algorithm

Step 1: Collect the data sets

Step 2: Apply Data filtration and transformation

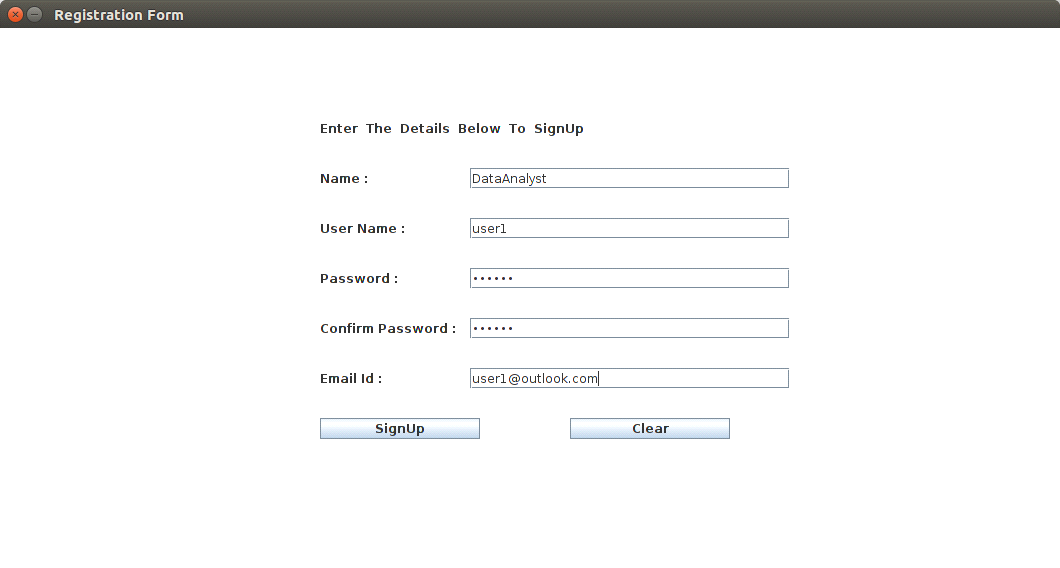
Step3: Upload Dataset to Hadoop Distributed file system

Step4: Perform Mining

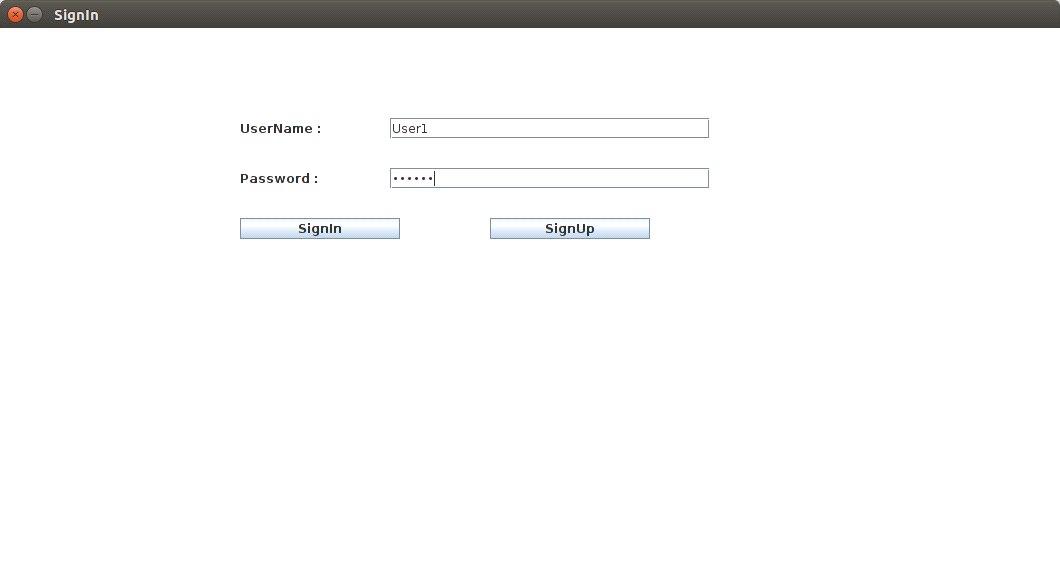
Step5: Download output Directory to desired location

Step6: Display Average Weighted support Vs Traditional support

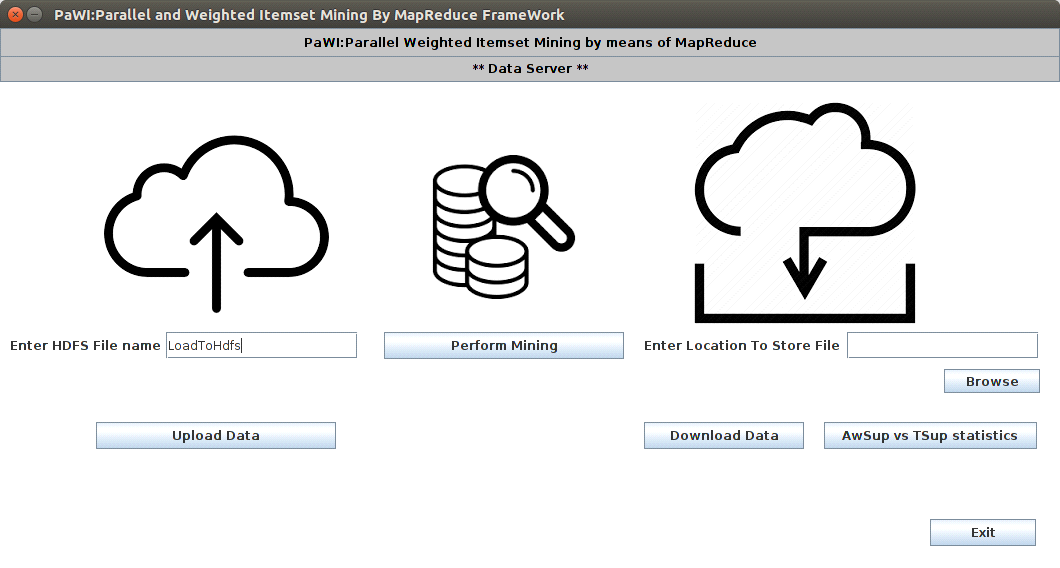
* **User Registration Module**

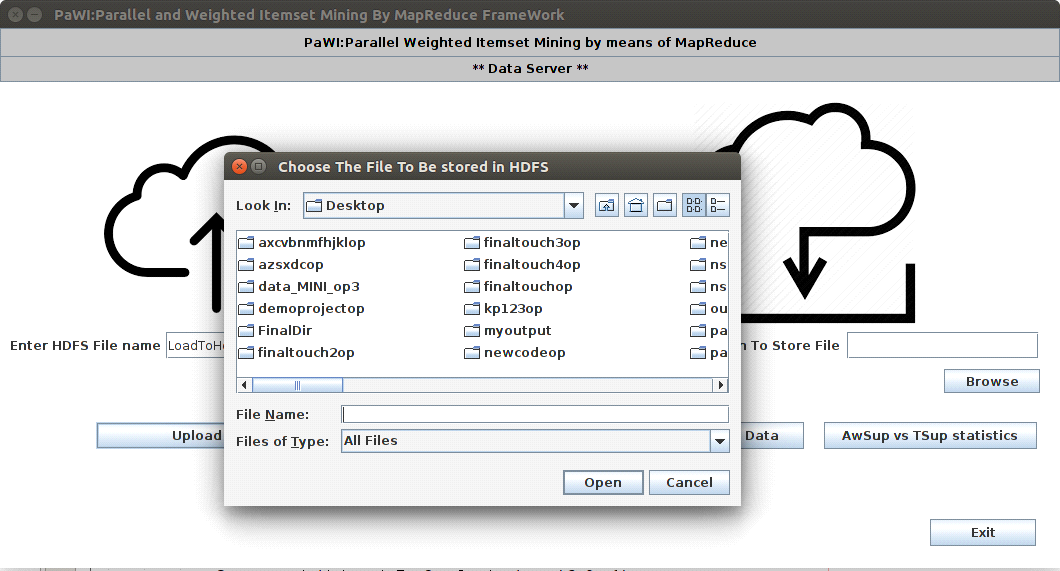


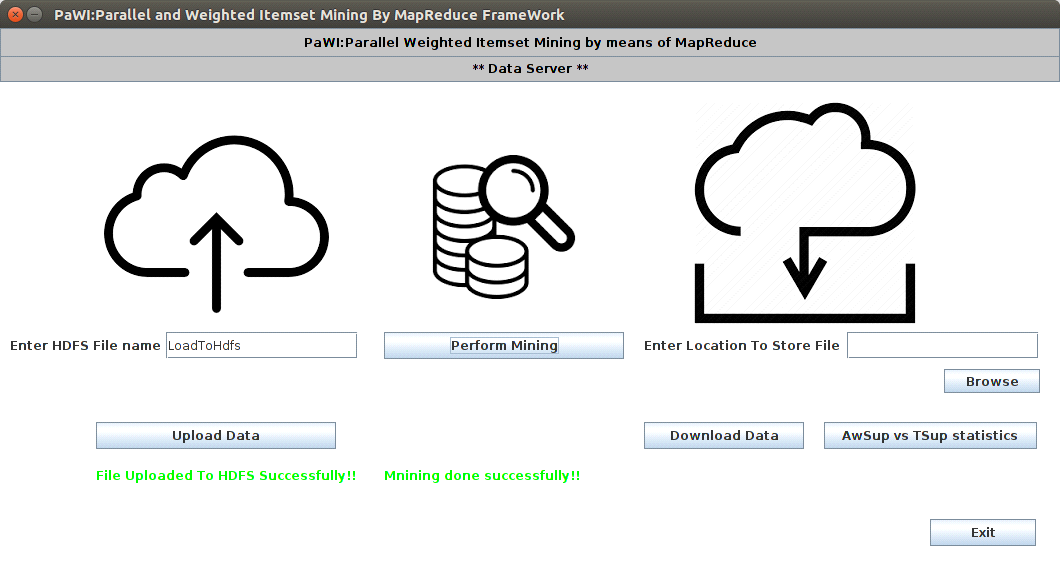
* **User SignIn Module**

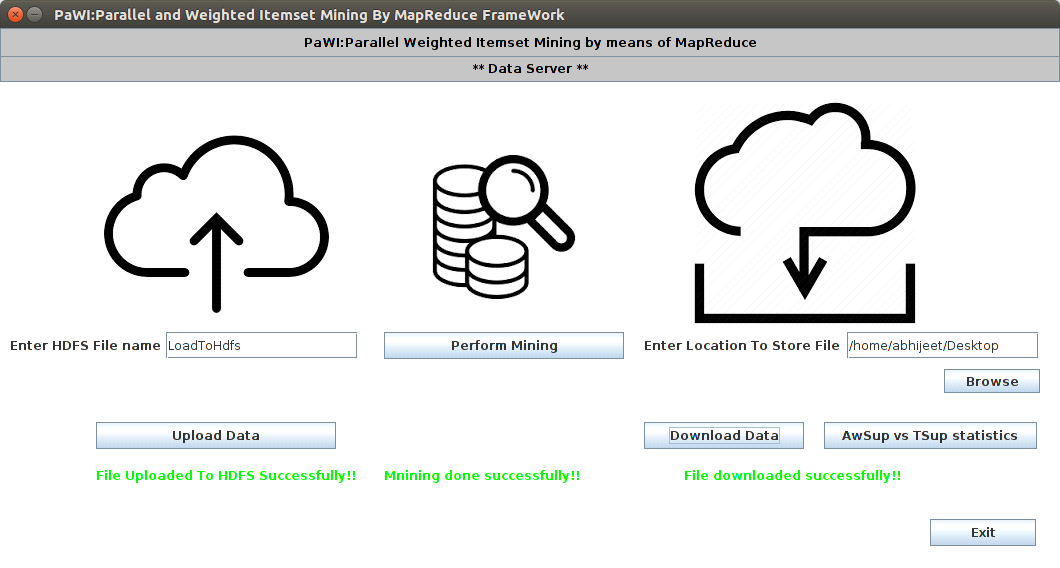


* **Data server Module**

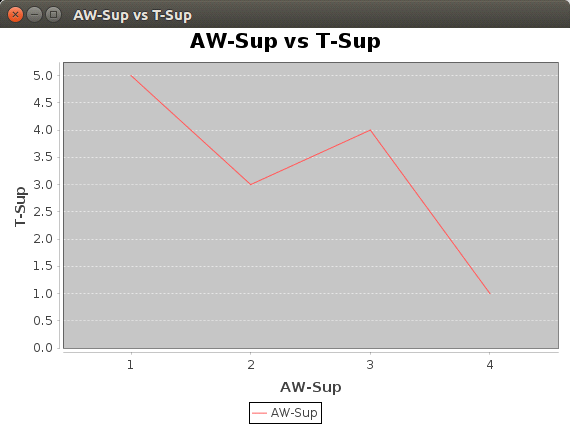








* **AwSup vs TSup Statistics**



**HARDWARE AND SOFTWARE USED**

**Hardware:** Following hardware were used for this project.

* **Processor intel core i5 – 5200U CPU @2.24GHZ**
* **RAM 4GB**

**Software:** Following software were used for this project.

* **Operating System :** UBUNTU 16.04 LTS
* **Apache Hadoop Framework 2.7.3**
* **NetBeans IDE 8.2**

**CONCLUSIONS AND FUTURE WORK**

This report presents a parallel and distributed solution to the problem of extracting frequent itemsets from Big Weighted Datasets. The proposed system, running on a Hadoop cluster, overcomes the limitations of approaches in coping with datasets enriched with item weights. The experiments, performed on a large volume of dataset, confirm the action ability of the mining result. Future works will entail the application of the proposed approach to recommender systems. For example, discovering combinations of items that were frequently bought together with an overall rating above average could be useful for recommending additional items beyond those already purchased by a given user.

**References**

* Elena Baralis, Luca Cagliero, Paolo Garza, “PaWI: Parallel Weighted Itemset Mining by means of MapReduce” [ 978-1-4673-7278-7/15 2015 IEEE DOI 10.1109/BigDataCongress.2015 ]
* R. Agrawal, T. Imielinski, and Swami, “Mining association rules between sets of items in large databases,” in ACM SIGMOD 1993, 1993, pp. 207–216.
* K. Sun and F. Bai, “Mining weighted association rules without preassigned weights,” IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 4, pp. 489 –495, 2008.
* S. Moens, E. Aksehirli, and B. Goethals, “Frequent itemset mining for big data,” in SML: BigData 2013 Workshop on Scalable Machine Learning. IEEE, 2013.
* M. Zaki, “Parallel and distributed association mining: a survey,” Concurrency, IEEE, vol. 7, no. 4, pp. 14–25, Oct 1999.
* R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in VLDB’94, Proceedings of 20th International Conference on Very Large Data Bases, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 1994, pp. 487–499.