Proposed Parallel Association Rules Algorithm

Dr. Emad kadhiem Jabbar
Depart of Computer Science, University of Technology/Baghdad
Email: emadalfatly@yahoo.com
Waheed Abd Al-Kadhiem Salman
Iraqi Commission for Computers & Informatics Informatics Institute for Postgraduate Studies, Computer Science Department

Received on: 20/3/2013 & Accepted on: 5/9/2013

ABSTRACT

Data mining is an advanced technique for extracting knowledge from a large amount of data for classification, prediction, estimation, clustering or association rules or any activities, which need decision. Mining for associations rules between items in large transactional distributed databases is a central problem in the field of knowledge discovery. When distributed databases are merged at single machine to mining knowledge it will require a large capacity of storage, long execution time in addition to transferring a huge volume of data over network might take extremely long time and also require an unbearable financial cost. In this paper an algorithm is presented toward saving communication cost over the network, central storage cost requirements, and accelerating required execution time. In this paper a new algorithm is proposed, called Proposed Parallel Association Rules Algorithm (PPARA) which aims to extract association rules from one record only for each site from distributed association rules in parallel instead of extracting association rules from huge quantity of distributed data at several sites in parallel, and that is through collecting the one record of local association rules from each site and storing it, these Local Association Rules turn in to produce global association rules over distributed systems in parallel.

Keywords: Data Mining, Association Rules, (PPARA) Algorithm, Local Association Rules, Global Association Rules.

خوارزمية قواعد الارتباط المتوازية المقترحه

الخلاصه

تنقيب البيانات هو تقنيه متقدمه لانتزاع المعرفه من كميه ضخمه من البيانات, للتصنيف والتوقع والتخمين والتجميع او لقواعد الارتباط او أي نشاطات, التي تحتاج الى قرار. ان تنقيب قواعد الارتباط بين العناصر في قواعد البيانات الصفقه الكبيره هي مشكله مركزيه في حقل اكتشاف المعرفه. عندما قواعد البيانات الموزعه تدمج في ماكنه واحده لتنقيب المعرفه الذي سيتطلب سعه كبيره من الخزن, وقت تنفيذ طويل بالاضافه الى ذلك تحويل حجم ضخم من البيانات عبر الشبكه قد يستغرق وقت طويل جدا ويتطلب ايظا كلفه ماليه لا تطاقفي هذه الورقةِ خوارزميةٌ مُقَدَّمةٌ باتجاه توفير كلفه الاتصال عبر الشبكه, ومتطلبات كلف الخزن المركزي , وتعجيل وقت التنفيذ المطلوب. في هذه الورقه خوارزميه جديده تدعى خوارزمية قواعد الارتباط المتوازيه المقترحه, الخوارزميه

التي تهدف لانتزاع قواعد الارتباط من سجل واحد فقط لكل موقع من قواعد الارتباط الموزعه بالتوازي بدلا من انتزاع قواعد الارتباط من الكميه الكبيره من البيانات الموزعه في عده مواقع بالتوازي وذلك خلال جمع سجل واحد من قواعد الارتباط المحليه من كل موقع وخزنهم, هذه قواعد الارتباط المحليه تحول لانتاج قواعد الارتباط العامه على الانظمه الموزعه بالتوازي.
خوارزمية قواعد الارتباط المتوازيه المقترحه

INTRODUCTION

With fast development in information technology, the role of computer and data systems has dramatically changed. Organizations need data systems not just to run the day-to-day business but also to help them in making strategic decisions. Decision-support systems have become commonplace in today's business environment. The extraction of useful and non-trivial information from the huge amount of data that is possible to collect in many and diverse fields of science, business and engineering, is called Data Mining (DM). DM is part of a bigger framework, referred to as Knowledge Discovery in Databases (KDD), which covers a complex process, from data preparation to knowledge modeling. Within this process, DM techniques and algorithms are the actual tools that analysts have at their disposal to find unknown patterns and correlation in the data. Typical DM tasks are classification, clustering, association rules, and others. Association rule mining is one of the most important and well researched techniques of data mining. The discovery of "association rules" in databases may provide useful background knowledge to decision support systems, selective marketing, financial forecast, medical diagnosis, and many other applications [1].

RELATED WORK

- In1996 Agra wall and Shaffer introduced the parallel algorithms based on count distribution and data distribution in order to solve the problems of data mining. Their structure is appropriate for data mining but regarding to the huge volume of data and the low speed of search in these algorithms, researchers have tried to increase their search speed [2].
- In 2001 Zaiane et al. proposed a parallel algorithm that is based on frequent pattern –grouth algorithm (fp-growth) .The algorithm is MLFPT (Multiple Local Frequent Pattern Tree). It assumes shared-memory architecture. Just like the centralized fp-growth algorithm, MLFPT does not generate candidates for frequent itemsets but instead builds multiple frequent pattern trees (FP-trees) [3].
- In 2010 Hussein Khidhr Abbas presented a new technique to find distributed association rules from distributed data mining distributed over distributed data warehouses .The proposed distributed association rules algorithm is called: Improving and Enhancing Distributed Association Rule Mining (IEDARM), It generates support counts of candidate itemsets more quickly than other DARM algorithms and reduces the size of average transactions, data sets, and message exchanges and does not require any distributed scan to the distributed sites to get the support values of the itemsets rather than all of existing distributed a priori-based approaches which require many scans of the distributed sites to get the value of the support [1].

ASSOCIATION RULES

Association rules are one of the promising aspects of data mining as knowledge discovery tool and have been widely explored to date, they allow capturing all

possible rules that explain the presence of some attributes according to the presence of other attributes [4]. An association rule is a rule, which implies certain association relationships among a set of objects, in a database. Given a set of transaction, where each transaction is a set of literal (called items), an association rule is an expression of the form X U Y, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions of the database, which contains X, tend to contain Y [5]. Association rules identify relationships between attributes and items in database such as the presence or absence of one pattern implies the presence or absence of another pattern. An association rule is an expression X→Y where X={x1, x2... xn} and Y={y1, y2... yn} are set of items with left hand side (LHS) and right hand side (RHS). The meaning of such rules is quite intuitive: given database (D) of transactions (T) where each transaction T Є D is a set of items, X→Y which expresses that whenever a transaction T contains X, the T probably contains Y. Also the probability of rule strength is defined as the percentage of transactions containing Y in addition to X. The prevalence of rule is the percentage of transactions that hold all the items in the union. If prevalence is low, it implies that there is no overwhelming evidence that items in X U Y occur together [6]. The important measures for association rules, support (S) and confidence (C) can be defined as: The support (S) of an association rule is the ratio (in percent) of the records that contain (X U Y) to the total number of records in database [7].

Support (X→Y) =P (XUY) (2-1) … (1)

Support (X→Y)=frequent(XUY)/total number of records in database..2-2

For given number of records, confidence (C) is the ratio (in percent) of the numbers of records that contain (X U Y), to the number of records that contain X. thus, if we say that a rule has a confidence of 85% it means that 85% of the records containing X also contain Y. The confidence of rule indicates the degree of correlation in the database between X and Y. Confidence is also a measure of rules strength [8].

Confidence (X→Y)=frequent(XUY)/frequent (2-3) …(2)

4- Proposed Algorithm: Proposed Parallel Association Rules Algorithm (PPARA).

The Proposed Algorithm is a new Algorithm which focuses on the principle of mining knowledge over geographical distributed systems in parallel computing. It attempts to get global association rules from locally distributed association rules in parallel implementation. The basic idea behind this proposed algorithm depends on finding global association rules from one record only of locally distributed association rules in parallel instead of extracting association rules from a large distributed data located at several sites. In other words; each site has responsibility to extract its own i.e. local association rules, and then construct a new record that contains all extracted association rules for each site. It depends on itemset relations with (k-1 itemsets and k+1 itemsets where k=2) and puts all these association rules in a controller site to find out the global association rules, which are more accurate than those mined from all raw data located at distributed sites when they are collected together. PPARA could play a significant task in distributed data mining

since it works with one record for each site instead of huge quantity of data records for each site.

The system consists of S branches (S1 ,S2 ,S3 ,…….etc) and consist of DB databases (DB1 ,DB2 ,DB3 ,…….etc) to extract association rules of each site such as (ARS1 ,ARS2 ,ARS3 ,……..etc). On the other hand, to extract the global association rules for the whole system requires (DB1+ DB2+ DB3) data records collected together at controller site . which needs large space of memory, long execution time and may cause losing some of vital association rules in its data site through collecting data, while PPARA algorithm extracts global association rules depending on distributed association rules (ARS1 ,ARS2 ,ARS3 ,……..etc) after receiving one record only for each site and collecting them together in controller site. So PPARA introduces good advantage to distributed database by isolating local analysis at each site from global analysis of association rules. And that implies reducing the communication overhead, central storage requirements, and computation times, as will be explained. The main flowchart is explained in Figure (1).

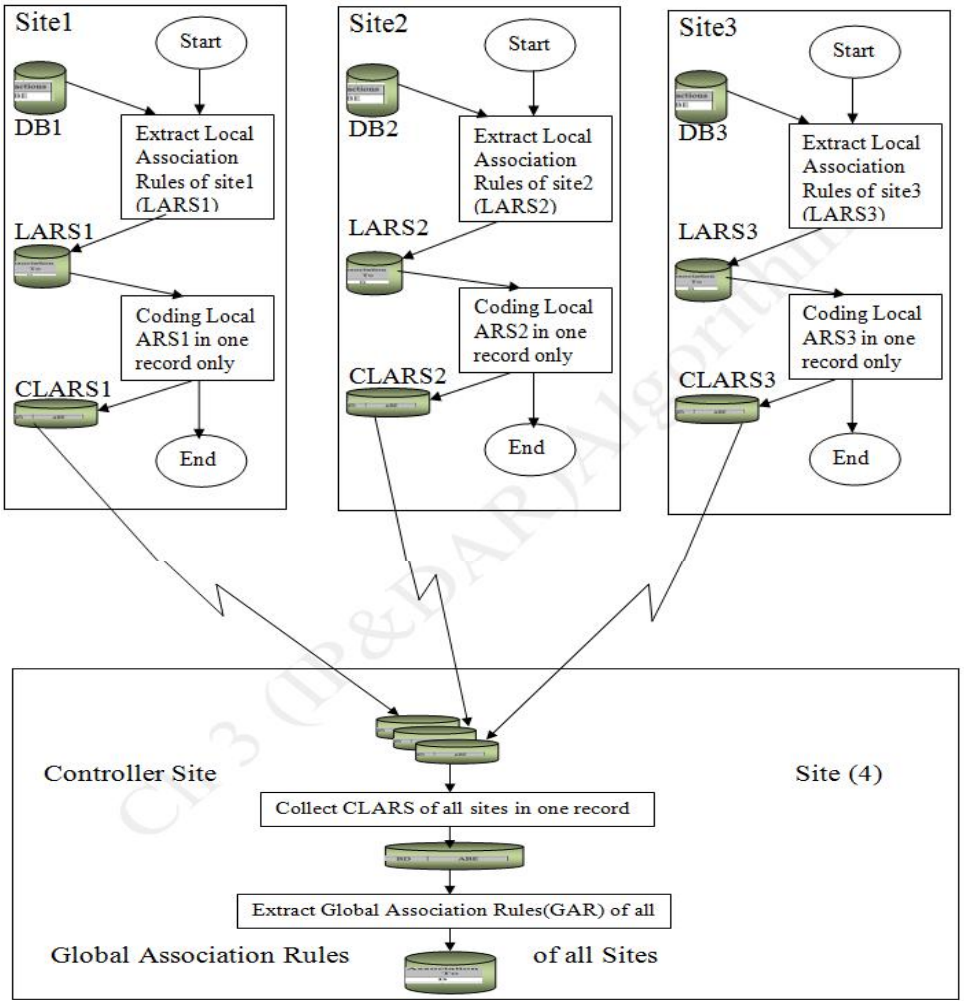

Figure (1) The main flowchart of PPARA Algorithm.

4-PPARA Algorithm Steps

The proposed algorithm consists of two phases:-

Phase one : Generate association rules for each site

Repeat all the following steps for each site

Step 1: Sort database according to item length and alphabet, then Reduce the number of records by collecting the same item records in one record with its frequency in new database table.

Step 2: Go to end of new database table, merge K-items record with nearest subset K-i items record, where K= max itemset and i = 1or 2 or 3 or ….n-1, and pointed them, where n is number of items.

Step 3: Convert merged table MT to binary coding database BCDB due to state merge with other records.

Step 4: Compute total frequency of each 1-itemset

Step 5: Construct k-itemsets table due to k > 1 and (frequent itemsets = > threshold) such as AB = A and B, ABC = A and B and C, ABCD = A and B and C and D …etc.

Step 6: Look for identical columns for k- itemsets = k-1 itemsets (where k=2) and their subset, If identical columns are found then that means existence of association rules.

Step 7: convert all Local association rules of site to one record only, then Send only one record of association rules for each site to controller site.

Phase two : Generate Global Association Rules (GGAR) for all sites in controller site as in :

Step 1: Controller site receives one record of association rules for each site.

Step 2: Collect the records of association rules for all sites in one record only by summing left site of value [L(Rk-1)] of each identical itemset and summing right site of value [R(Rk+1)] of each identical itemset in each site and append the other itemsets.

Step 3: Generate Global Association Rules for all system.

AN APPLICABLE EXAMPLE

Example: Let's look at a system which has three branches distributed at three different sites S1,S2, S3, and S4 is company center for controlling the three sites and giving reports to the higher management to make decisions by extracting global association rules from distributed association rules in parallel as shown in Table (1).

Table (1) Databases of three sites.

site1 (S1)

TID	List of item IDs
T1	ABE
T2	D
T3	ABCDE
T4	BC
T5	ABD
T6	C
T7	BC
T8	C
T9	ABCD

site2 (S2)

TID	List of item IDs
T1	ABC
T2	ABE
T3	D
T4	BC
T5	ABD
T6	BC

site3 (S3)

TID	List of item
T1	AE
T2	AE
T3	BCD
T4	ABE
T5	CD
T6	CD

Solution: Now to get Association Rules at control site (S4), there are two techniques:

Traditional techniques (all transactions of sites).

This technique uses Apriori algorithm to compute global association rules from raw data for all sites and the results are shown in Table (2).

Note the results have only one association rule.

Table (2) Global Association Rules by Apriori.

Association rules
E ⟶ A

PPARA Technique

Phase one : Generate association rules for each site

Step 1: Sort database according to item length and alphabet, then Reduce the number of records by collecting the same item records in one record with its frequency in new database Table, as shown in Table (3).

Table (3) Sorting DB of site 1.

TID	C. Tran.	Count
T6,T8	C	2
T2	D	1
T4,T7	BC	2
T5	ABD	1
T1	ABE	1
T9	ABCD	1
T3	ABCDE	1

Step 2: Go to end of Table (3), merge K-items record(ABCDE) with nearest K-i items record(ABCD) when K-i items record is subset from K-items record, and point them, where K=2.

§ Put K-i items (ABCD) and frequency of them (K-items record and K-i items record) from their records in fields K-i items merge, F1 respectively, as in Table (4).

§ Put differences of K-items and K-i items merge (E) where (ABCDE – ABCD =E) and frequency of K-items record in fields Diff. of Merge, F2 respectively, as in Table (4).

§ Repeat this step until beginning of file for each unpointed records.

(Merged table) Table (4).

ID	K-i items Merge	F1	Diff. of Merge	F2
1-	ABE	1	---	0
2-	D	2	AB	1
3-	C	4	B	2
4-	ABCD	2	E	1

F1 = frequency of K-items record + frequency of K-i items record
F2 = frequency of K-items record
Diff. of Merge = difference of merge = K-items record - K-i items record
= ABCDE – ABCD = E

Step 3 : Prepare Table of 1-itemset which consists of fields for each item (A,B,C,D,E) and its frequency and then convert Merged table MT to binary coding database BCDB due to state merge with other records. From table (4)

§ IF item i such as (A) is not found in record j then put (00) binary code to item i, else IF item i such as (A) is found in Diff. of merge of record j then put (01) binary code to item i, else IF item i such as (A) is found in K-i items record and not found value in Diff. of Merge then put (01) binary code to item i, else IF item i such as (A) is found in K-i items record and value is found in Diff. of Merge then put (11) binary code to item i.

§ Put F2 value of Table (4) into F2 field of Table (5).

§ Compute F3 value of Table (5) by (F3 = F1 - F2) of Table (4).

§ Repeat this step for each record in Table (4).

Step 4: Compute total frequency of each 1-itemset

§ IF binary code (01) & F2 = 0 then F/= F + F2

§ IF binary code (01) & F2 = 0 then F = F + F3

§ Else IF binary code (11) then F = F + [F2 + F3] as given in Table (5).

Table (5) Binary code of 1-itemset.

TID	A	B	C	D	E	F2	F3
1-	01	01	00	00	01	1	0
2-	01	01	00	11	00	1	1
3-	00	01	11	00	00	2	2
4-	11	11	11	11	01	1	1
Total	4	6	6	4	2		

Step 5: Construct k-itemsets table due to k > 1 and (frequent itemsets = > threshold) such as AB = A and B, ABC = A and B and C, ABCD = A and B and C and D …etc.

Table (5) Binary code 2-itemset.

TID	AB	AC	AD	AE	BC	BD	BE	CD	CE	DE	F2	F3
1-	01	00	00	01	00	00	01	00	00	00	1	0
2-	01	00	01	00	00	01	00	00	00	00	1	1
3-	00	00	00	00	01	00	00	00	00	00	2	2
4-	11	11	11	01	11	11	01	11	01	01	1	1
Total	4	2	3	2	4	3	2	2	1	1		

Table (5) Binary code 3-itemset.

TID	**ABC**	**ABD**	**ABE**	**ACD**	**BCD**	**F2**	**F3**
1-	00	00	01	00	00	1	0
2-	00	00	00	00	00	1	1
3-	00	01	00	00	00	2	2
4-	11	11	01	11	11	1	1
Total	2	3	2	2	2		

Table (5) Binary code 4-itemset.

TID	ABCD	F2	F3
1-	00	1	0
2-	00	2	2
3-	00	1	1
4-	11	1	1
Total	2		

Table (5) Binary code 5-itemset.

TID	ABCDE	F2	F3
1-	00	1	0
2-	00	2	2
3-	00	1	1
4-	01	1	1
Total	1		

Step 6: Look for identical columns in Table (5) for k- itemsets = k-1 itemsets (where k=2) and subset of it, If identical columns are found then that means existence of association rules as listed in Table (6).

Table (6) Local Association Rules of Site 1(LARS1).

No	1-itemset columns	2-itemset
1	A	AB
2	E	AE
3	E	BE
No	2-itemset columns	3-itemset
1	AC	ABC
2	AC	ACD
3	AD	ABD
4	AE	ABE
5	BD	ABD
6	BE	ABE
7	CD	ACD
8	CD	BCD
No	3-itemset columns	4-itemset
1	ABC	ABCD
2	ACD	ABCD
3	BCD	ABCD
No	4-itemset columns	5-itemset
	No identical columns	No identical

Association	Association
A	B
E	A
E	B
AC	B
AC	D
AD	B
AE	B
BD	A
BE	A
CD	A
CD	B
ABC	D
ACD	B
BCD	A

Step 7: By using Table ((6) Site 1) convert all Local association rules of site to one record only by :-

§ Putting each K-itemset of identical columns with K-1 itemset or K+1 itemset without duplicating as in Table (7)

§ Computting frequency of each of them, with K-1itemset and putting the frequent itemset on the left side.

§ Computting frequency of each of them ,with K+1itemset and putting the frequent itemset on the right side.

§ L = represents number of relations to this itemsets with k-1 itemsets L(Rk-1).

§ R = represents number of relations to this itemsets with k+1 itemsets R(Rk+1) , as inTable (7).

§ Send only one record of association rules for each site to controll site.

Table (7) Coding Local Association Rules of Site 1(CLARS1).

A		E		AB		AC		AD		AE		BD		BE		CD		ABC		ABD		ABE		ACD		BCD		ABCD	
0	1	0	2	1	0	0	2	0	1	1	1	0	1	1	1	0	2	1	1	2	0	2	0	2	1	1	1	3	0

Left side of value = number of relation with k-1 itemset L(Rk-1)
Right side of value = number of relation with k+1 itemset R(Rk+1)

Phase two: Generate Global Association Rules (GGAR) for all sites in controller site (site4)

Step 1: Controller site receives one record of association rules for each site as shown in Table (8).

Step 2: Collect the records of association rules for all sites in one record only by summing left site of value [L(Rk-1)] of each identical itemset and summing right site of value [R(Rk+1)] of each identical itemset in each site and append the other itemsets, as shown in Table(8).

Table (8) Collect (CLARi) of all sites in controller site.

A		E		AB		AC		AD		AE		BD		BE		CD		ABC		ABD		ABE		ACD		BCD		ABCD	
0	1	0	2	1	0	0	2	0	1	1	1	0	1	1	1	0	2	1	1	2	0	2	0	2	1	1	1	3	0

One record of Site 1 (CLARS1)

A		C		AB		BC	
0	1	0	1	1	0	1	0

One record of Site 2 (CLARS2)

A		C		D		E		AE		CD	
0	1	0	1	0	1	0	1	2	0	2	0

One record of Site 3 (CLARS3)

A		C		D		E		AB		AC		AD		AE		BC		BD		BE		CD		ABC		ABD		ABE		ACD		BCD		ABCD	
0	3	0	2	0	1	0	3	2	0	0	2	0	1	3	1	1	0	0	1	1	1	2	2	1	1	2	0	2	0	2	1	1	1	3	0

One record of all sites (CLARS all)

Step 3: Generate Global Association Rules for all system as shown in Table (9).

Table (9) Global Association Rules for all sites.

ID	Association From	Association To
1	A	B
2	E	B
3	E	A
4	C	B
5	A	E

6	D	C
7	C	D
8	AE	B
9	BE	A
10	AC	B
11	AC	D
12	AD	B
13	BD	A
14	CD	A
15	CD	B
16	ABC	D
17	ACD	B
18	BCD	A

COMPARISON BETWEEN PPARA AND TRADITIONAL TECHNIQUES

This section makes a comparison between PPARA and traditional techniques to measure efficiency of the proposed algorithm. They are implemented in site4 (the controller site), and (traditional techniques) applied directly by using a A-priori algorithm on raw data of all sites and then (PPARA) indirectly through local association rules (one record only) of sites. Then Global Association Rules results are compared as shown in Figure (2) and execution time charts are compared as shown in Figure (3).

Table (10) gives details about the power of proposed algorithm when compared with that of traditional technique in its two approaches (on all raw data and on all association rules from each site) covering a number of transactions works, execution time and storage required space, and finally the number of association rules results which are named or called the Association Rules (AR), as well as other vectors.

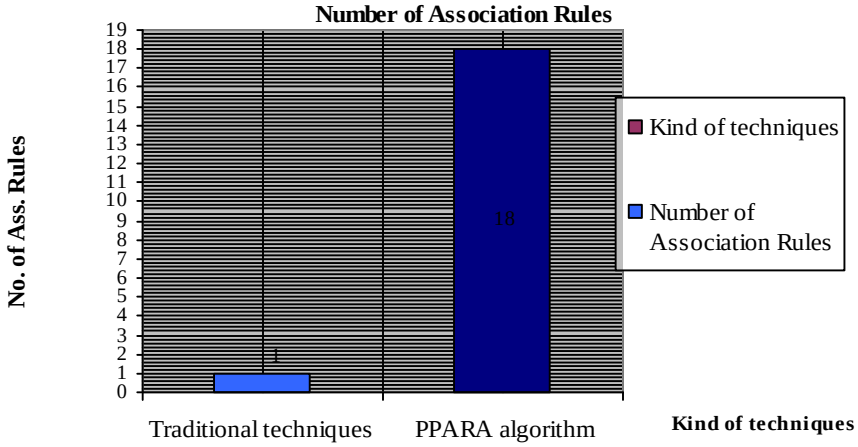

Figure (2) Comparison between result Association Rules obtained from two implemented methods.

Table (10) Table of implementations compared.

Compared vectors	Apriori with all Raw Data	PPARA algorithm
Transactions number	4,500,000	3
Execution time (second)	1200	1.24
Storage space	25 MB	0.0125 KB
Global Ass. Rules No.	1	18
Data applying	Directly	Indirectly
Loss Ass. Rules per site	Loss of many important Ass. Rules per site	Not loss

ACCURACY OF RESULTS

The PPARA was implemented to find the global association rules which are more accurate than the global association rules which were found from all of the raw data by using traditional technique, since PPARA guarantees correct and independent local analysis for each site. That is because it's keeping the private data at each site and works its association rules which are computed locally at its own site, and then the global association rules are mined from it.

Storage Cost

PPARA works with one record only (which is basically local association rules record for each site) instead of huge quantity of records. Therefore, PPARA will reduce required storage sized.

Communication Cost

Transferring a huge volume of data over network might take extremely much time and also requires an unbearable financial cost. PPARA saves time and money needed because it works on the distributed association rules (one record) of each site instead of using the raw data of all sites.

Execution Time

PPARA needs less execution time because it works with local association rules (one record only) instead of all raw data. In other words PPARA works with one record only (which is basically local association rules record for each site) instead of a huge quantity of records.

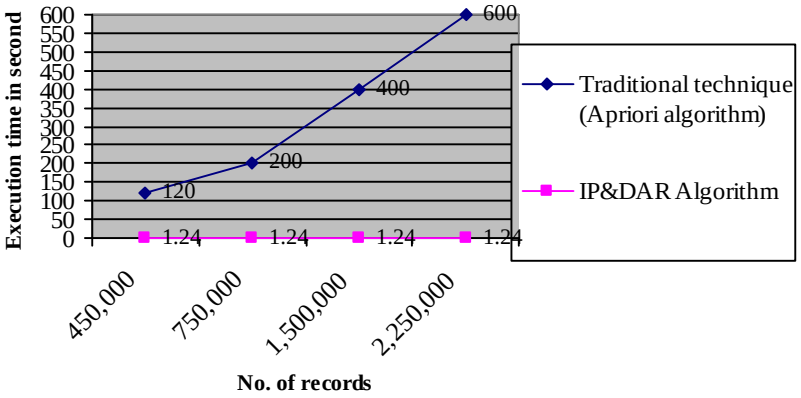

Figure (3) Comparison between execution time for two Implemented methods.

CONCLUSIONS

We have introduced and discussed the use of PPARA in the task of discovering association rules in distributed database in parallel.

The conclusions which are drawn from implementing the proposed algorithm in real world and comparing its results with those that are obtained from the most famous traditional technique i.e. A-priori are:-

1. Applying proposed algorithm doesn't require huge quantity of transmutation data and that will reduce size of storage in controller site and through network.
2. The huge data is reduced to one record of association rules for each site.
3. High performance in extracting association rules is carried out throughreducing execution time and storage space.
4. Using AND logic operation makes it convincing to get 100% of relation out of the relation ratio that is required to compute the confidence.
5. Extracting association rules from association rules gives the optimal case of the relations between sites.
6. PPARA algorithm reduces Communication cost. Since the transfer of huge data volumes over network might take extremely long time and also requires an unbearable financial cost. This is avoided by the PPARA algorithm. Also the algorithm utilizes the network resources by minimizing message transfer among sites.
7. PPARA algorithm solves the problem of true negative and false positive association rules which appear in some DARM algorithms in order to collect local association rules to generate global ones.
8. Many processes in many sites are used to extract association rules.
9. Also threshold isn't required with proposed algorithm.
10. The compressed database can be decompressed to the original form.
11. I/O time is reduced by using only the compressed database to do data mining.
12. Incremental data mining is allowed.

REFERENCES

[1]. Hussein K. Abbas," Algorithm of Association Rules in Distributed Data Mining". Ph.D. Thesis Computer Science, University of Technology, 2010.

[2]. Agrawal, R. and J.C. Shafer, 1996. "Parallel mining of association rules". IEEE Transactions on Knowledge and Data Engineering, 8(6): 962-969.

[3]. Osmar R. Zaiane, Mohammad El-Hajj, and Paul Lu. "Fast Parallel Association Rule Mining without Candidacy Generation". In ICDM, pages 665–668, 2001.

[4]. Charu C. Agrawal and Philip S. Ya, March 1998, "Mining large itemsets for association

[5]. Pieter Adriaans, Dolf Zantinge, 1998, "Data Mining", Addision Wesley.

[6]. Rakesh Agrawal, Heikki Mannila, Srikant R., Hannu Toivenen and A. Inkeri Verkamo, 1996, "Fast discovery of association rules", Springer publisher, Santiago de Chile.

[7]. Sergy Brin, Rajeev Motwani, Jeffery D. Ullman and Sergy Tsur, May 1997, "Dynamic itemset counting and implication rules for market basket data", proceeding of data (SGMOD97) Tucson, Arizona USA.

[8]. Chen M. S., J Han and P.S. YU, 1996, "Data mining an overview from a database perspective", IEEE trans, knowledge and data Engineering.

[9]. Maimon Oded, Rokach Lior, 2005, "The Data Mining and Knowledge Discovery Handbook", Springer, USA.