Intern. J. Fuzzy Mathematical Archive
Vol. 5, No. 2, 2014, 57-69
ISSN: 2320 –3242 (P), 2320 –3250 (online)
Published on 20 December 2014
www.researchmathsci.org

International Journal of Fuzzy Mathematical Archive

An improved mechanism of clustering the sites for Peer-to-Peer Distributed Databases

***D.I.George Amalarethinam*[1] and *C.Balakrishnan*[2]**

[1]Department of Computer Science, Jamal Mohamed College (Autonomous)
Tiruchirappalli, India. e-mail: di_george@ymail.com
Corresponding Author

[2]Software Development, Skill Development Centre, Alagappa University, Karaikudi
India. e-mail: balasjc@gmail.com

Received 3 December 2014; accepted 19 December 2014

Abstract. The design aspect of distributed database environment is a major research issue. With the characteristics like, robustness and ability to scale, the Peer-to-Peer Distributed Database architecture has the potential to handle the data in an efficient manner. This work proposed an improved methodology to cluster the sites based on locality reference value for Peer-to-Peer architecture, to address the issues in fragmentation and allocation phases of database design. This work takes the inspiration of the previous works done based on the predicate based fragmentation and introduces the clustering approach for drafting the database architecture and to allocate the fragmented data across the sites.

Keywords: Peer-to-Peer databases, fragmentation, allocation, priority factor values, clustering approach

1. Introduction

Distributed processing of data is an efficient way of improving the performance of Database Management Systems (DBMSs) and applications that manipulate large volumes of data. A distributed database is a collection of multiple, logically interrelated databases distributed over a computer network [1]. This resource distribution improves performance, reliability, availability and modularity that are inherent in distributed systems. There will be a possibility of improved response times to queries and upgrading system capacity or performance incrementally. Design of Distributed database environment is one of the major research issues in the area of distributed database system.

The conceptually simplest distribution scheme is to distribute at the table level: any given table is stored in its entirety at some site or it may be partitioned and stored in different sites. A technique of breaking up the database into logical units, which may be assigned for storage at the various sites called data fragmentation. Fragmentation can be horizontal, vertical and mixed or hybrid. Allocation describes the process of assigning each fragment or each copy of a fragment to a particular site in the distributed system. Fragment placement strategies may be centralized, partitioned or fragmented and replicated. The replication of fragments helps to improve availability, performance of retrieval of global queries and reliability.

Peer-to-Peer (P2P) technology has no strict definition; it is generally described as having a structure that is contrast to the traditional client-server model. Each node in the network acts as both client and server, requesting data from neighboring nodes as well as routing and serving data for others. The nature of P2P technology makes it well suited for storing multiple copies of data between several nodes, in turn offering reliable access to data and distributing the load of requests. Additionally, the multiple links between nodes make the system more stable as nodes are asdded and dropped. All the features inherent in P2P technology promise a network that is dynamic, scalable and reliable.

In this paper, cluster based architecture of the distributed databases to address the fragmentation and allocation phases of database design has been introduced. This work takes the inspiration of the previous works done based on the predicate based fragmentation and introduces the clustering approach for drafting the database architecture and for allocating the data across the sites.

The paper is organized as follows. The next section of this work presents literature reviews of grouping the sites, fragmentation, allocation and clustering. Section III describes the clustering approach based on locality reference value and initial allocation. The performance of proposed clustering methodology is compared with Chord structure. Finally Section IV concludes the paper with future research directions.

2. Literature review

This section of the paper states the related works that are stimulated to do research on methodology for fragmentation and allocation of data over multiple sites of the network. Most of the research related to fragmentation and allocation has been carried out in the context of relational databases. Navathe [2] has proposed a mixed fragmentation method. It is based on a graph theoretic algorithm which clusters a set of attributes and predicates into a set of vertical and horizontal fragments, respectively. Horizontal fragmentation algorithm for distributed deductive database systems has been proposed by Lim et. Al [3]. This algorithm handled the horizontal fragmentation by clustering all the tuples in a base relation that are used by queries. Lim and Yiu-Kai Ng [4] presented different approaches for vertical fragmentation of relations and allocation of rules and fragments. It helps to maximize locality of query evaluation and minimizes communication cost and execution time during processing the queries. Huang and Chen [5] proposed a simple and comprehensive model for a fragment allocation problem. Also, they have developed Huang and Chen, two heuristics algorithms to find an optimal allocation of the fragments. Ahmad et al., [6] have addressed the allocation of fragments problem in distributed database system. They have developed a query driven data allocation approach. Various algorithms based on evolutionary computing paradigm have also been proposed by them. Du et al., [7] have proposed new algorithms based on a new measurement to evaluate togetherness among the attributes in a relation. Hababeh et al., [8] proposed a method for allocating fragments to a cluster. Sites in the distributed database systems are grouped based on their communication cost. A method for incrementally maintaining the primary horizontal fragments of an object oriented database has been proposed by Campan et al., [9]. Abdalla and Marir [10] made a comparative study on vertical partitioning algorithms to find the most efficient vertical partitioning schema. Hui Ma and Markus Kirchberg

[11] presented a cost-based approach for horizontal and vertical fragmentation. Algorithms were presented for each of the fragmentation techniques used in distribution design to obtain fragmentation schema, which would improve the system performance. Abuelyaman [12] proposed a vertical partitioning algorithm for improving the performance of database systems without the knowledge of empirical data. The algorithm uses the number of occurrences of an attribute in a set of queries rather than the frequencies of queries accessing these attributes. Singh and Kahlon [13] proposed a new dynamic data allocation algorithm for non-replicated distributed database system. Khan and Hoque [14] have proposed a new technique of fragmentation to solve the problem of taking fragmentation decision at the initial stage of a distributed database design, according to the attribute locality precedence table. Dimovski et al., [15] presented a novel formal approach for horizontal partitioning of relations based on predicate abstraction. This paper proposes an approach for clustering the sites for PPDDBS. The clustering process is done based on the locality reference value of the particular site. This locality reference value of a site enables to determine whether or not a set of sites is assigned to a certain cluster. This clustering approach considered as a fast way to determine the data allocation to a set of sites rather than site by site, the Horizontal fragmentation technique is adopted for fragmentation. The priority factor values [14] of the attributes of a relation is considered as the criteria for fragmentation. This priority factor value of the attribute of a relation is derived by constructing the *Enhanced Create, Update, Read and Delete* (ECURD) matrix [14]. The initial allocation is done based on the preliminary assumption of priority factor value of the relations. The following sections explain the approach for the methodology of clustering the sites for fragmentation and process of initial allocation of fragments to the sites.

3. An improved approach for clustering of sites

The locality reference value of each site is considered for clustering process. A site in the network will possess a Site Information Table (SIT), which contains information regarding site ID and Region. Here, Site ID refers the unique identification number of the site and Region refers the locality division of the site in the network. Ten sites are considered to explain the clustering process as shown in Figure 1

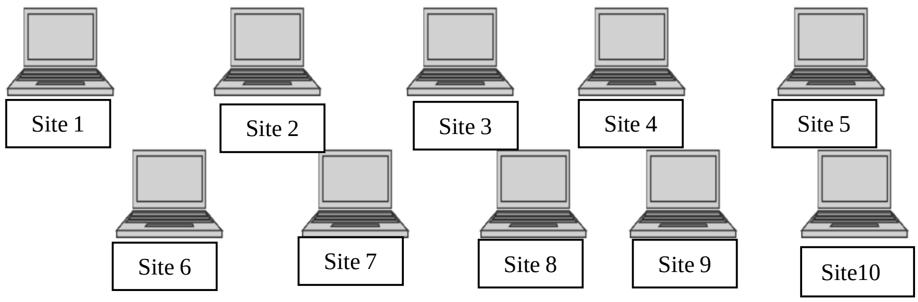

Figure 1: Sites taken for consideration

The attributes of SIT of ten sites are given in Table 1. The clustering process is done at the top level (architecture level) based on the number of regions of the sites using SIT. Thus, the sites are clustered with its locality reference value. The grouping of sites to each Cluster are given in Table 2

Site Id	Region
S1	R1
S2	R2
S3	R1
S4	R3
S5	R2
S6	R1
S7	R3
S8	R4
S9	R4
S10	R1

Table 1: Site Information Table

Site Id	Region	Cluster
S1	R1	Cluster1
S3	R1	Cluster1
S6	R1	Cluster1
S10	R1	Cluster1
S2	R2	Cluster2
S5	R2	Cluster2
S4	R3	Cluster3
S7	R3	Cluster3
S8	R4	Cluster4
S9	R4	Cluster4

Table 2: Classification of sites to the respective Clusters

It is noted from the Table 1 that there are four regions, namely, R1, R2, R3 and R4. The number of clusters is equal to the number of regions. Hence, there will be four clusters, namely, Cluster 1, Cluster 2, Cluster 3 and Cluster 4. The region-wise grouping of sites to the four Clusters are given in Figure 2

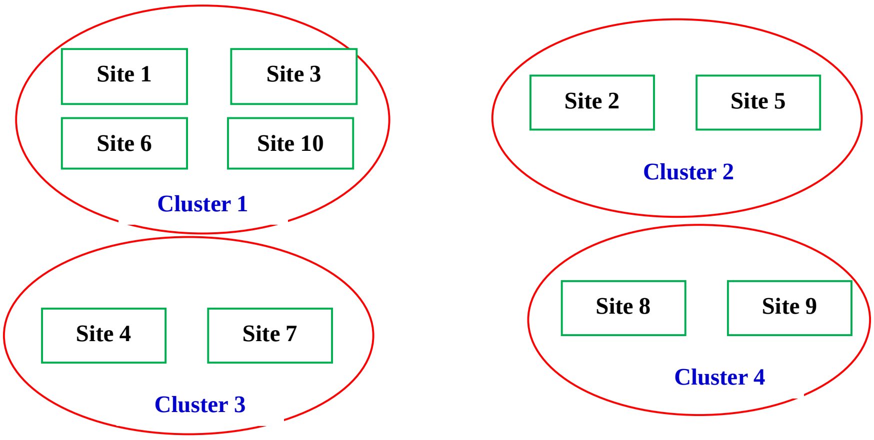

Figure 2: Clustering of sites based on Locality reference value

When the clustering process is over, the process of fragmentation and allocation take place. The following section describes the derivation of ECRUD matrix to get priority factor of each attribute to enhance the Horizontal fragmentation and allocation of fragments to the sites.

3.1 Constructing of ECRUD matrix for fragmentation

Among the three fragmentation techniques, such as, Horizontal, Vertical and mixed fragmentation, this paper uses Horizontal fragmentation technique. From the literature review, the Horizontal fragmentation had the following problems [1] in common:

- They use frequency of queries, minterm predicates' affinity or attribute affinity matrix (AAM) as a basis of fragmentation. These methods require sufficient empirical data that are not available in most cases at the initial stage.

- Most of them have concentrated only fragmentation problem and overlooked allocation problem to reduce complexity.

In order to overcome these problems, the fragmentation process with Horizontal fragmentation approach can be done based on the locality priority factor of attributes in the relation. In the technique described by Shahidul Islam Khan and Dr. A. S. M. Latiful Hoque [14], the cost is treated as the effort of access and modifications in calculating the Attribute Locality Priority value. This paper considers the communication cost along with access and modification costs. The efforts made to perform operations such as, Create, Read, Update and Delete on relations from a particular site is considered for constructing the ECRUD matrix. Based on the values in ECRUD matrix an Attribute Locality Priority Table (ALPrT) is constructed. The horizontal fragmentation of relation is done by considering highest priority factor attribute of the relation. The sub-relations will be given to the clusters. Within the cluster, the repeated number of re-fragmentations is performed by considering the subsequent highest priority factor attributes. The repeated re-fragmentation process continued until the number of sub-relations that are equal to the number of sites in the cluster. The flow diagram explicitly describes the enhanced technique for fragmentation and allocation which is given in Figure 3.

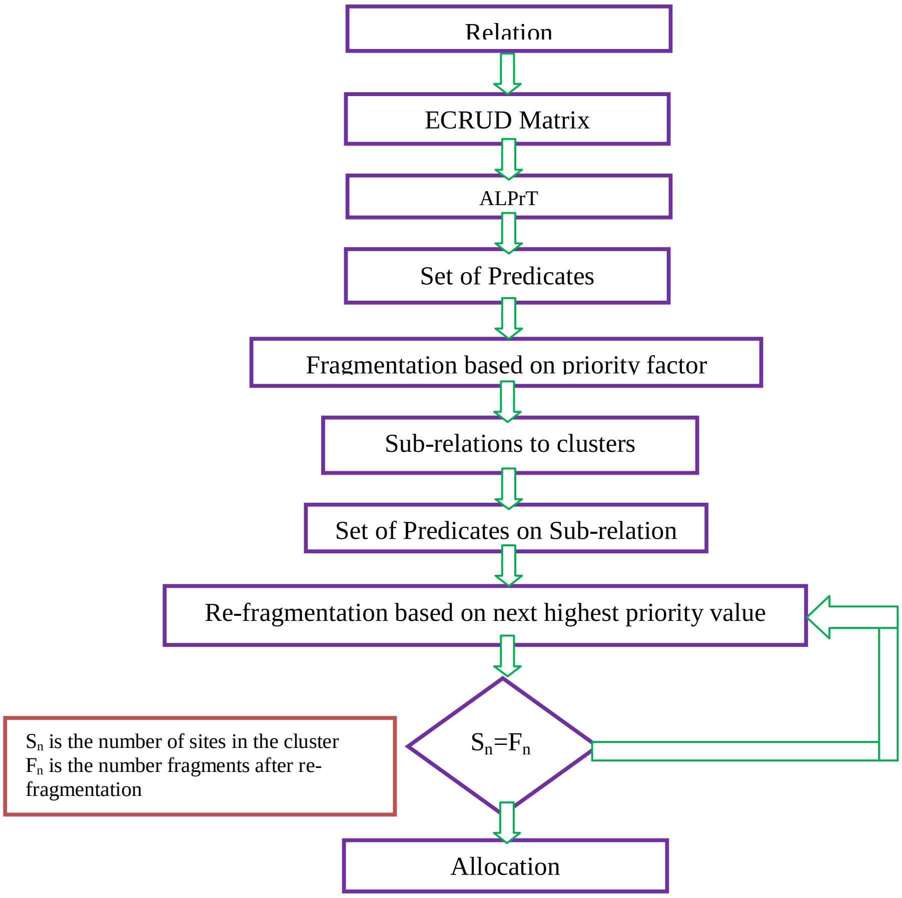

Figure 3: Flow diagram of the process of Fragmentation and Allocation

The cost is treated as the effort of access, modification of a particular attribute of a relation by an application from a site and average communication cost among sites. The ECRUD matrix for a relation is constructed by using the following cost functions. The Priority factor of an attribute of a relation can be calculated with the help of ECRUD matrix.

$$C_{x,i,z,y} = f_C\, C + f_R\, R + f_U\, U + f_D\, D$$

$$\mathrm{Tx, i, z} = \sum_{y=1}^{Ax,i,z} \mathrm{Cx, i, z, y}$$

$$M_{x,i,m} = \mathrm{Max(Tx, i, z)}$$

$$\mathrm{APx, i} = \mathrm{Mx, i, m} - \sum_{z \neq m}^{Ax,i,z} \mathrm{Tx, i, z}$$

$$\mathrm{APx} = \sum_{i=1}^{k} \mathrm{APx, i} + \mathrm{CC} \qquad (1)$$

where,

f_C = frequency of create operation
f_R = frequency of read operation
f_D = frequency of delete operation
f_U = frequency of update operation
C = weight of create operation
R = weight of read operation
D = weight of delete operation
U = weight of update operation
$C_{x,i,z,y}$ = cost of predicate i of attribute x accessed by application y at site z.
$T_{x, i, z}$ = sum of all applications' cost of predicate i of attribute x at site z.
$M_{x,i,m}$ = maximum cost among the sites for predicate i of attribute x.
$AP_{x, i}$ = actual cost for predicate i of attribute x.
AP_x = total cost of attribute x.
CC = average communication cost among the sites.

The actual frequencies of read, write, delete and update of a particular attribute from different applications of a site is un-known, because the fragmentation is done at the initial stage.

Hence, it is assumed that f_C, f_R, f_D and f_U =1 and C=2, R=1, D=2 and U=3. The reason is that at the design time of a distributed database, the designer is not aware of the actual frequencies of read, delete, create and update of a particular attribute from different applications of a site. In general from the past history, the update operation requires more cost than create and delete operations, also reading operation needs least cost.

The Attribute Locality Priority Table (ALPrT) is formed with the help of ECRUD matrix. A predicate set is generated by considering the order of the priority factor values of the attributes of a relation.

Each relation is fragmented horizontally by considering the highest priority factor attribute from the predicate set. After the clustering process, the sub-relation will be given to the Cluster. The re-fragmentation is done based on the next highest priority factor attribute from the predicate set.

The allocation of fragments to the sites takes place only when the number of sub-relations equals to the number of sites in the cluster. Hence, the repeated number of re-fragmentations is performed by considering the subsequent highest priority factor attributes.

3.2. Experiments for analyzing fragmentation and initial allocation

For analyzing the above narrated fragmentation and initial allocation algorithm, this research considers an account relation with the attributes as shown in Table 3

Accno	Category	Cid	Date	Balance	Region
1	A	C1	11/1/14	21000	R1
2	B	C2	21/1/14	13500	R2
3	B	C3	2/2/14	18000	R1
4	C	C4	8/2/14	22000	R3
5	D	C5	24/2/14	3200	R4
6	C	C6	15/3/14	52000	R1
7	E	C7	18/3/14	38000	R2
8	D	C8	28/3/14	12500	R1
9	A	C9	4/4/14	16800	R3
10	A	C10	9/4/14	78000	R1
11	B	C11	11/4/14	23000	R4
12	B	C12	18/4/14	11800	R2

Table 3: Account relation

The number of sites and clusters are considered as indicated in Figure 1 and Figure 2 respectively. The *ECRUD* matrix is constructed for the Account relation during the requirement analysis phase. From this matrix ALPrT values can be calculated using the cost functions mentioned in Equation 1 as stated in the previous section 3.1.

A part of *ECRUD* matrix of the Account relation (A) for the attribute category (C) is shown in Table 4

S.A	S1		S2		S3		S4		S5		S6		S7		S8		S9		S10	
E.A.P	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2
A.C =R1	CRUD	CRUD			R						R							R		
A.C =R2							CRUD	CRUD												
A.C =R3	CRUD	CRUD							R					R						R
A.C =R4																	CRUD	CRUD		

Table 4: A part of ECRUD matrix of Account Relation

3.2.1. Calculating attribute locality priority factor (ALPr)

Attribute Locality Priority (ALPr) factor values of each attribute will be calculated from the *ECRUD* matrix of the Account relation using the cost functions as indicated in Equation (1). The value of the predicate R1 of attribute Region is calculated as follows,

(1) Cost of predicate R1 of attribute Region accessed by application 1 at site 1 is 8 and for application 2 at site 1 is 8. It is calculated using the formula as stated in Equation 1,

$$C_{x,i,z,y} = f_C\, C + f_R\, R + f_U\, U + f_D\, D$$

For example, for R1, the ALPr values calculation is shown below,

S.A / E.A.P	S1		S2		S3		S4		S5		S6		S7		S8		S9		S10	
	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2
A.C =R1	**CRUD**	**CRUD**			**R**						**R**							**R**		

Table 5: ECRUD matrix values for calculation of ALPr of 'Region R1'

For A1 in S1: 2+1+3+2 => 8
For A2 in S1: 2+1+3+2 => 8
For A1 in S3: 1
For A1 in S6: 1
For A2 in S9: 1

In the same manner the value is calculated for all the sites.

(2) Sum of all applications' cost of predicate R1 of attribute Region at site 1 is 16, at sites 3, 6 and 9 is 1. It is calculated using the formula

$$\text{Tx, i, z} = \sum_{y=1}^{Ax,i,z} \text{Cx, i, z, y}$$

For R1:
For S1: A1+A2
8+8 => 16
For S3: A1 => 1
For S6: A1 => 1
For S9: A1 => 1

(3) Maximum cost among the sites for predicate R1 of attribute Region is 16. It is calculated using the formula,

$$M_{x,i,m} = \text{Max(Tx, i, z)}$$

For R1:
Maximum cost among the sites in R1 is 16 (for S1)

(4) Actual cost for predicate R1 of attribute Region is 13, R2 is 16, R3 is 13 and R4 is 16. It is calculated using the formula,

$$\text{APx, i} = \text{Mx, i, m} - \sum_{z \neq m}^{Ax,i,z} \text{Tx, i, z}$$

For R1:
Actual cost = S1 cost – (S3+S6+S9)
= 16 – (1+1+1) = 13

(5) Attribute locality Priority factor of attribute Region is (13+16+13+16) = 58. Total cost of attribute Region is calculated using the formula,

$$\text{APx} = \sum_{i=1}^{k} \text{APx, i}$$

In the same manner Priority factor values of all the attributes can be calculated using the *ECRUD* matrix.

From the ALPr values of all the attributes, the Attribute Locality Priority Table (ALPrT) for the Account relation is constructed and shown in Table 6.

Name of Attribute	Priority factor Value
ACCNO	10
CATEGORY	25
CID	11
DATE	14
BALANCE	18
REGION	58

Table 6: ALPrT of Account relation

The highest Priority factor valued attribute will be considered as a key attribute for fragmentation. According to that predicate set will be generated. For instance, from Table 6, ALPrT shows that Region has the highest Priority factor value. So the predicate set will be as follows:

P = {Region=R1; Region=R3; Region=R2; Region=R4}

Based on these predicate sets, relation will be fragmented. The relational algebraic notations for the fragmentation and re-fragmentation of relations are as follows in the Equations 2, 3 and 4.

Let

R	be the Relation
n	be the number of sites
t_i	are the tuples of the relation
t_h	is set of ordered tuples based on highest priority factor value
t_{hl}	is a tuple having highest priority factor value
t_{hn}	is a tuple having next highest priority factor value
t_{hm}	is a tuple having next highest priority factor value
SR_i	be the Sub-relations of original relation for Clusters
RSR_i	be the re-fragmented relations of sub-relation for the sites within the Cluster

NSR_i be the next level re-fragmented relations of re-fragmented relations for the newly added sites of the Cluster

Equation 2 is used for Fragmentation of relation into sub-relations for Clusters

$$SR_i \rightarrow \sigma\ t_{hx}\ ^{(R)} \text{ where } i = 1\ldots n \text{ and } x = 1 \quad (2)$$

By applying Equation 2, the fragments for clusters are attained as stated in Table 7

ANO	CATEGORY	CID	DATE	BALANCE	REGION
Predicate is 'Region=R1'					
1	A	C1	11/1/14	21000	R1
3	B	C3	2/2/14	18000	R1
6	C	C6	15/3/14	52000	R1
8	D	C8	28/3/14	11500	R1
10	A	C10	9/4/14	78000	R1
Predicate is 'Region=R2'					
2	B	C2	21/1/14	13500	R2
7	E	C7	18/3/14	38000	R2
12	B	C12	18/4/14	11800	R2
Predicate is 'Region=R3'					
4	C	C4	8/2/14	22000	R3
9	A	C9	4/4/14	16800	R3
Predicate is 'Region=R4'					
5	D	C5	24/2/14	3200	R4
11	B	C11	11/4/14	23000	R4

Table 7: Sub-relation based on the predicates

Equation 3 will be used for Fragmentation of sub-relation into re-fragmented relations for sites

$$RSRi \rightarrow \sigma\ thy\ (SR) \text{ where } i = 1\ldots n \text{ and } y = 2 \quad (3)$$

Equation 4 indicates the selection methodology for fragmenting the re-fragmented relations into sub-relations for newly added sites or if the number of sites is more than the number of re-fragmented relations

$$NSR_i \rightarrow \sigma\ t_{hz}\ ^{(RSR)} \text{ where } i = 1\ldots n \text{ and } z = 3\ldots n \quad (4)$$

The re-fragmentation takes place by applying equation 3 and 4 on the Table 7. The results of re-fragmentation and the allocation of fragments into the sites of the respective clusters are depicted in the Table 8. If another site is added to any of the clusters, next highest Priority factor valued attribute will be taken for further fragmentation. Thus, the initial allocation process done on the sites of the Cluster 1 is indicated in Table 8

SITE ID	FRAGMENT ID
S1	f1
S3	f2
S6	f3
S10	f4

Table 8: Initial allocation for the sites in Cluster 1

3.3. Performance analysis

The performance of proposed clustering methodology has been analyzed with Chord structure. Since the Chord structure is a very successful implementation in Peer-to-Peer information sharing systems. The simulated experimental results are given in the following Table 9:

S.No	Number of Fragments & Sites	Execution time in Chord (millisecond)	Execution time in proposed clustered structure (millisecond)
1	20	285	276
2	25	362	277
3	30	432	276
4	35	528	277
5	40	561	277

Table 9: Experimental results to test the Scalability of proposed clustering methodology

The results displayed in Table 9 clearly explained the scalability of proposed clustering methodology. The results show the ability of the proposed clustering methodology to produce consistency upon the various experimentations ranges from 20 sites to 40 sites. The Figure 4 clearly explained the scalability nature of proposed clustering methodology.

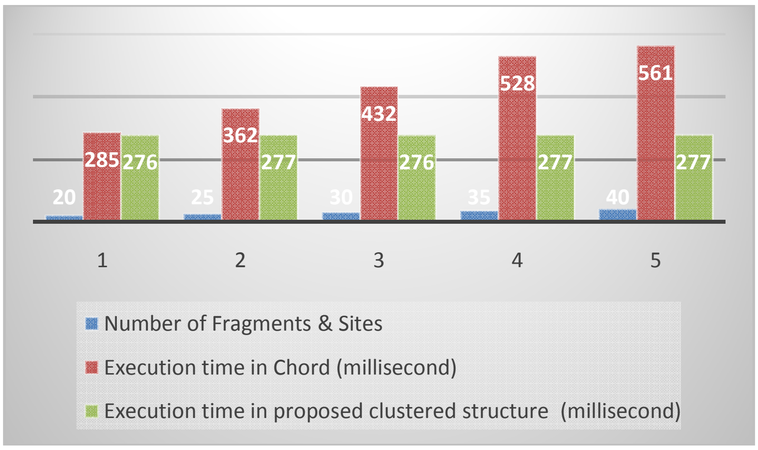

Figure 4: Scalability of proposed clustering methodology

4. Conclusion

This paper focused on proposing an improved approach for clustering the sites based on the locality reference value. The findings for clustering process are tabulated and shown in figures. The ECRUD matrix is created by considering the cost for the Create, Read, Update and Delete operations on the attributes of the relation. The Attribute Locality Priority Table (ALPrT) for a relation is formulated with the help of ECRUD matrix. A predicate set is formed by ordering the attributes based on the priority factor value derived by using ALPrT. The Horizontal fragmentation of relation is done by considering the highest priority factor attribute from the ALPrT. The sub-relation is given to the respective cluster. Based on the next highest priority value attribute of the sub-relation, the re-fragmentation is carried out to create multiple numbers of sub-relations. The allocating the sub-relations to the sites of the cluster is done only when the number of re-fragmented sub-relations equals to the number of sites in the cluster. Otherwise, the re-fragmented sub-relations are once again horizontally fragmented based on the next highest priority value in the ALPrT. The process of fragmentation and initial allocation are stated and explained with examples of Account relation. The performance of this proposed clustering methodology has been studied with Chord structure the results were satisfactory.

The future work may analyze the optimized data re-allocation strategies into the sites of the cluster.

REFERENCES

1. M.T.Ozsu and P.Valduriez, *Principles of Distributed Database Systems*, 2nd ed., New Jersey: Prentice-Hall, 1999.
2. S.Navathe, A Mixed Fragmentation Methodology for Initial Distributed Database Design, 1995.
3. Seung-Jin Lim and Yiu-Kai Ng, A formal approach for horizontal fragmentation in distributed deductive database design, *Proceedings of the 7th International Conference on Database and Expert Systems Applications*, (1996).
4. Seung-Jin Lim and Yiu-Kai Ng, Vertical fragmentation and allocation in distributed deductive database systems, *Information Systems*, 22(1) (1997) 1-24.
5. Yin-Fu Huang and Jyh-Her Chen, Fragment allocation in distributed database design, *Journal of Information Science and Engineering,* 17 (2001) 491-506.
6. I.Ahmad, Y.-K.Kwok and S.-K.So, Evolutionary algorithms for allocating data in distributed database systems, *Distributed and Parallel Databases,* 11(1) (2002) 5-32.
7. Jun Du, Ken Barker and Reda Alhajj, Attraction- a global affinity measure for database vertical partitioning, *Proceedings of ICWI*, (2003).
8. I.O.Hababeh, A method for fragment allocation design in the distributed database systems, *The Sixth Annual U.A.E. University Research Conference*, (2005).
9. Alina Campan, Sergiu Adrian Darabant and Gabriela Serban, Clustering techniques for adaptive horizontal fragmentation in object oriented databases, *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics ICTAMI,* pp 263-274, (2005).
10. Hassan I. Abdalla and F. Marir, Vertical partitioning impact on performance and manageability of distributed database systems (A comparative study of some vertical partitioning algorithms), *18th National Computer Conference, Saudi Computer Society*, (2006).
11. Hui Ma and Markus Kirchberg, Cost-based fragmentation for distributed complex value databases, *Proceedings of the 26th International Conference on Conceptual modeling*, (2007).
12. Eltayeb Salih Abuelyaman, An optimized scheme for vertical partitioning of a distributed database, *IJCSNS International Journal of Computer Science and Network Security*, 8(1) (2008) 310-316.

13. Arjan Singh and K.S. Kahlon, Non-replicated dynamic data allocation in distributed database systems, *IJCSNS International Journal of Computer Science and Network Security,* 9(9) (2009) 17-180.
14. Shahidul Islam Khan and A. S. M. Latiful Hoque, A new technique for database fragmentation in distributed systems, *International Journal of Computer Application,* 5(9) (2010).
15. A.Dimovski, G.Velinov and D.Sahpaski, Horizontal partitioning by predicate abstraction and its application to data warehouse design, *Proceedings of the 14th east European conference on advances in databases and information systems,* (2010).