Volume 8, No. 5, May-June 2017

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Clustered Approach to Dynamic Data Allocation in Distributed Database Systems

Raju Kumar
Research Scholar
Department of Computer Science
GurukulKangriVishwavidyalaya, Haridwar India

Neena Gupta
Assistant Professor
Department of Computer Science
KanyaGurukulMahavidyalayaDehradun(Second Campus of Gurukul Kangri Vishwavidyalaya, Haridwar),India

Abstract: Distributed database is very much popular in organizations which are distributed and operating from multiple geographic locations. The two important issues to improve the performance of distributed database systems (DDS) are - grouping of different sites into clusters, and dynamic reallocation of data fragment to these clusters. Clustering network sites and data allocation are still open research problems as their optimal solution is NP-hard. The prime contribution in this field is to develop a near optimal solution. In this paper a new algorithm is proposed for clustering distributed database sites based on the distance between sites to minimise the number of communications and network overhead between sites. Moreover, a Clustered Approach to Dynamic Data Allocation (CADDA) algorithm is proposed to dynamically reallocate data fragments to clusters in redundant and non-redundant distributed database system to reduce the remote data accesses and network overhead. The proposed approach is implemented on a sample distributed database system and compared with other cluster and non-cluster based data allocation algorithms. The comparison results show that proposed approach efficiency is better and it improves the overall performance of the distributed database system.

Keywords: Distributed database system, Redundant and non-redundant, Dynamic data allocation, Cluster of sites, DDS performance.

1. INTRODUCTION

The recent advances in database technology and network technology caused distributed database system (DDS) a favourable choice for geographically dispersed organizations. DDS is considered as a collection of logically interrelated data that are actually allocated at several locations over a computer network [1]. Each site of the network can perform local applications independently. Each site also must contribute to the execution of at least one global application, which requires data accessing at different sites through a communication subsystem [2]. Although distributed database is in great demand, before adopting, its several security issues and challenges need to be considered carefully [3].

To improve distributed database performance several methods have been proposed. Improvement in the performance can be gained by improving one or more of the following database management issues: database fragmentation, data allocation & replication, and clustering of different sites. The task of fragment allocation falls under NP-hard problem so its complexity is high [1]. To solve such problems heuristic algorithms are appropriate approach. To achieve the data availability, system reliability, and system performance, each fragment can be allocated to one or more than one network nodes.

In clustering approach distributed database sites are grouped into logical clusters to reduce the number of extra communications and associated costs between the network sites which results as enhanced DDS performance. Static data allocation is used in almost all cluster based approaches developed for improving the DDS performance. In this paper a new Clustered Approach to Dynamic Data Allocation (CADDA) algorithm is proposed, which dynamically reallocates data among different clusters. This algorithm easily handles the dynamic data reallocation case, when at the same time more than one clusters qualify for data reallocation in redundant and non-redundant distributed database system. The comparison between proposed approach and other approaches proposed in papers [4,5,6,7,8,9]proves that the new proposed approach is more efficient and significantly enhances the performance of the system.

In [10]a threshold algorithm for non-replicated distributed databases was introduced, which relocates data fragments as per the changing data access patterns and data access threshold value. In[11] an algorithm called Threshold and Time Constraint Algorithm (TTCA) was proposed, where the relocation of non-redundant data was performed according to the changing data access patterns,data access threshold value and time constraint in distributed database systems. Scaling problem was the main drawback with this algorithm. To remove the scaling problem of TTCA, an Extended Threshold Algorithm (ETA) was proposed in [12], which also minimized space requirements. A new algorithm was introduced in [7]which dynamically reallocates data in non-redundant distributed database system based on access threshold, time constraints of database accesses and volume of data transmission. Thereafter[8] extended the work done in [7]by additionally introducing distance parameter which enabled the algorithm for efficiently handling the situation where multiple sites qualify for fragment relocation.

To allocate data dynamically in redundant distributed database system, [13] introduced an algorithmwhich was based on fragment's correlation, lazy replication strategy, and non-uniform distances between network sites. Authors of [14, 15] proposed a dynamic fragment re-allocation model,which re-allocates data across sites on basis of update and communication cost values for each fragment individually. For dynamic data allocation in redundant and non-redundant distributed database system, a systematic survey of 31 research papers was presentedin [16]. In [9] an efficient algorithm was proposed, which extended the work

carried out in [7, 8], this algorithm also reallocates data dynamically in redundant distributed database system.

Many database researchers proposed the clustering method for data allocation to minimize number of communications and associated communication cost. In[4]an integrated method for clustering and fragment allocation in distributed database was proposed. Its clustering method reduced the number of communications and associated communication costs among the sites. In [5] clustering method was used to improve the performance of distributed database system by improving transactions response time.To improve the performance of distributed database system, a new integrated approach was proposed in [6] by combining three enhanced techniques - database fragmentation, network sites clustering and data allocation. Cluster based data allocation methods presented in [4,5,6] used static data allocation.

The remaining sections of the paper is presented as follows: In section 2 algorithm for clustering network sites is described. Section 3 demonstrates the cluster performance. Section 4 describes the proposed CADDA algorithm for data allocation in redundant and non-redundant distributed database system. Section 5 demonstrates algorithm working with sample database. In section 6 the comparison of proposed CADDA algorithm with algorithms proposed in [4,5,6,7,8,9] is performed**.**At last in section 7, the conclusion of the study is presented.

2. CLUSTERING NETWORK SITES

Suppose that distributed database has eight sites connected with one another with some communication links. Each site can directly communicate with any other site as shown in figure 1.

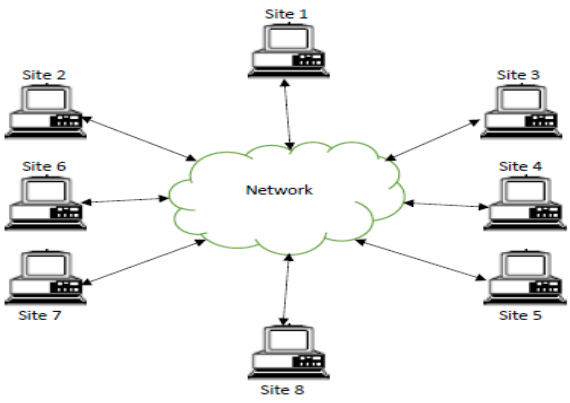

Figure 1: Distributed Database System with Eight Sites

The parameters considered for the proposed clustering technique are described as follows:

• *Logical Cluster C_i*

Logical place that used to group network sites together based on some physical property like distance between them exist.

• *Distributed Network Sites*

Set of fully connected network sites S_i, S_j , . . . S_n, of distributed database system. Each site is the place from where the transactions are triggered, and transaction results are held.

• *Distance Range DR*

The maximum distance value (in Km) that is allowed between the DDS network sites for grouping into the same cluster can be decided by the network administrator. Shortest path method is used to calculate the distance between two sites.

• *Distance $D(S_i, S_j)$*

The shortest path distance between two sites $S_i$and $S_j$in the DDS.

• *Cluster Site Matrix CSM*

Calculated matrix by which the clusters are created and their network sites are assigned.

• *Clustering Decision Value CDV*

The binary value that determineswhether a pair of sites $S_i$and $S_j$can be grouped together in the same cluster. CDV is calculated using following formula:

$$CDV(Si, Sj) = \begin{cases} 1 & : \text{if } D(Si, Sj) <= DR \wedge i \neq j \\ 0 & : \text{if } D(Si, Sj) > DR \vee i = j \end{cases}$$

It is obvious that CDV for the same site is equal to zero. If the $CDV(S_i, S_j)$ is equal to 1, then sites S_i, $S_j$are grouped into the same cluster, otherwise they are assigned to different clusters. Suppose the eight sites of distributed database are placed at some distance (in Km) from one another according to the site distance matrix shown in table 1.

Table 1. Site Distance Matrix

Site→ ↓	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1	0	90	600	680	400	700	900	800
S_2	90	0	700	800	500	400	550	750
S_3	600	700	0	80	50	300	500	700
S_4	680	800	80	0	100	400	600	800
S_5	400	500	50	100	0	650	700	850
S_6	700	400	300	400	650	0	85	95
S_7	900	550	500	600	700	85	0	60
S_8	800	750	700	800	850	95	60	0

For setting up an efficient clustering method, it is assumed that each site is assigned to only one cluster. The clustering algorithm is described as follows:

Clustering algorithm

Input:

Site Distance Matrix showing distance between sites $D(S_i, S_j)$
Distance Range (DR)
Number of network sites of distributed database system (NS)

Processing

/*Determining the sites that match the distance range in order to group them in one cluster*/
Step 1: Set 1 to i
Step 2: Do steps (3–12) until i>NS
Step 3: Set 1 to j
Set 0 to cluster site matrix CSM
Step 4: Do steps (5–10) until j >NS
Step 5: If i ≠j AND $D(S_i, S_j)$ <= DR, go to step (6)
Else, go to step (7)
Step 6: Set 1 to the $CSM(S_i, S_j)$

Go to step 8
Step 7: Set 0 to the CSM(S_i, S_j)
Step 8: End IF
Step 9: Add 1 to j
Step 10: Loop
Step 11: Add 1 to i
Step 12: Loop

Output:

Cluster Site Matrix (CSM) having generated clusters and their respective network sites

End

Suppose distance range (DR) value is 100 Km. After using the above clustering algorithm and site distance matrix, a cluster site matrix (CSM) is produced as shown in table 2.

Table 2. Cluster Site Matrix

Site → Cluster↓	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
C_1	1	1	0	0	0	0	0	0
C_2	0	0	1	1	1	0	0	0
C_3	0	0	0	0	0	1	1	1

After clustering, the resulting distributed database system is shown in figure 2.

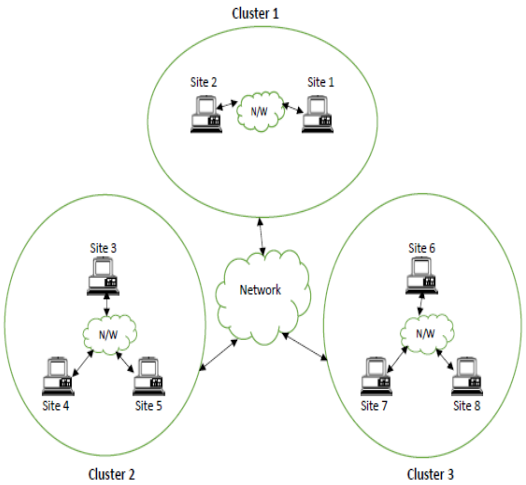

Figure 2: Distributed Database System after Clustering

3. PERFORMANCE EVALUATION OF CLUSTERING

In this section the details about the performance improvement obtained by proposed clustering algorithm is evaluated. For sake of simplicity, it is assumed that the sites as well as clusters of distributed database system are fully connected as shown in figure 2. One site act as a cluster head in each cluster. If the cluster head fails, next node within the cluster will act as cluster head; although it will come at certain cost but will reduce the downtime and increase the availability. Network/Database administrator has the flexibility to decide a cluster head/next cluster head. Generally, the site which is at the central position with respect to others sites within the cluster can be considered as cluster head. Communications among clusters can be performed through cluster heads.

Number of communications: Generally grouping of sites into clusters reduces the number of communications. The figure 1 showing distributed database of 8 sites, each site is communicated with the other 7 sites, so the initial total number of communications is (8 X 7 = 56). After clustering the sites into three clusters, each cluster is communicated with the other 2 clusters, so number of communications between clusters is (3 X 2 = 6) and taking into account the communication within the cluster itself; in this case the total number of communications is 20. Therefore, a high-performance is achieved by using proposed clustering method, which reduces the number of communications from 56 to 20 and enhances the system progress by 64.29 % as calculated using the following formula:

Improvement percentage = (*Reduced number of communications/Total number of communications) X100*

Improvement percentage = (36 / 56) X 100 = 64.29 %

4. PROPOSED CADDA ALGORITHM

Initially data fragments are allocated to distributed database system sites using any static data allocation algorithm. Each fragment may be allocated at one or more sites. Thereafter using proposed clustering algorithm, the sites are grouped into disjoint clusters. The proposed clustered approach to dynamic data allocation (CADDA) algorithm is described in two phases- preparation phase and action phase. The description about various notations used throughout the paper are shown in table 3.

Table 3. Algorithm Notations

Notation	Meaning
X	Number of data fragments in distributed database system
Y	Number of sites in distributed database system
Z	Number of clusters in distributed database system
F_i	The i^{th} data fragment
S_j	The j^{th} site
C_k	The k^{th} cluster
α	Access threshold for fragment relocation
β	Time constraint for fragment relocation
A_p^q	Access log record for p^{th} access at cluster q
n_i^j	Total number of accesses from cluster C_j to the fragment F_i within time interval β up to current access time t
V_i^j	Volume of data transmitted between fragment F_i and cluster C_j within time interval β up to current access time t
D_i^j	Distance between cluster C_i and cluster C_j

Preparation Phase:

Consider a distributed database system having Y sites and X data fragments. These all sites are grouped into Z clusters

using proposed clustering algorithm. In each cluster there is a site which acts as a cluster head and responsible for cluster to cluster communication. The interaction within the sites of a cluster as well as among the clusters are performed through some communication network. Each site has one or more fragment allocated to it. The site fragment allocation matrix is assigned to each site. The cluster site allocation matrix is stored at each cluster, so that it can prepare its access plan. Each cluster also keeps the value of two parameters- Access Threshold for fragment reallocation (α) and Time constraint for fragment reallocation (β).

Dynamic reallocation of data fragments within the sites of a cluster can be left out. Since the sites within a cluster are at closer distance, so relocating data fragments within the cluster sites will not reflect a significant improvement in system performance. Moreover, dynamic reallocation of data fragments among the sites of a cluster can be allowed on system administrator's wish. In that case each site within the cluster has to additionally store site distance matrix, site-access-log table and the value of Access Threshold for fragment reallocation (α1) and Time constraint for fragment reallocation (β1) within the cluster.The value of α and α1 as well as β and β1 may be the same or different.Now data reallocation within the sites of a cluster can be performed using approach discussed in [9]. At each cluster various activities are performed as per following steps:

*Step1:*Allocate all the data fragments to different sites of distributed database using any static data allocation method in replicated / non-replicated manner.

*Step2:*At each cluster set the constant value for Access threshold (α) and Time constraint (β).

*Step3:*At each cluster store a row of cluster distance matrix, which shows the cluster distance from all other clusters.

*Step4:*At each cluster store Access_Log table having following schema:

Access_Log (AFID, ACID, ADateTime, DataVol)

Where AFID represents ID of the accessed fragment, ACID represents ID of the cluster which accesses the fragment, ADateTime represents date and time of fragment access, and DataVol represents volume of data transmitted to and from the accessed fragment. An access log record is stored at each cluster for each access to the fragments allocated to that cluster. $\mathbf{A_p^q}$ represents an Access_Log record which means p^{th} access at cluster C_q, where p=1,2,3,...∞ and q = 1,2,3,….,Z.

Step5: At any particular time t (say 24 hours i.e. 00:00:00 - hh:mm:ss) access Access_Log table daily at each cluster and delete all records which is older than time constraint β up to current access time t.

For example, if time interval β up to current access time t is 10 days. This may be implemented at each cluster by executing a SQL query like "Delete from Access_Log where DATEDIFF (SYSDATE,ADateTime) >10".

Each transaction must preserves the ACID (Atomicity, Concurrency, Isolation and Durability) properties, whereas [17] supports for ACIA (Atomicity, Concurrency, Isolation and Availability) properties by giving preference to high availability over durability.Further it is assumed that there exists some distributed concurrency control mechanism that maintains ACID/ACIA properties for all transactions; and ensures that for every write operation performed on a fragment which is stored at a particular site of a cluster, changes made to the fragment should be reflected to all clusters/sites wherever copy of that fragment is stored.

Action Phase:

This phase involves a set of activities, which are performed during each fragment access. Suppose at time t, cluster C_j accessesfragment $F_i$which is initially allocated at cluster C_q, where i= 1,2,3,...X, j=1,2,3,.....,Z, q= 1,2,3,...,Z, and q = j or q≠ j. At cluster C_q the following steps 1 to 8 activities are performed by local agent for every access to a fragment stored at this cluster by some application or query invoked from the different or same cluster:

Step1: For every access made to fragment F_i allocated at cluster C_q, write a log record A_p^qin Access_Log table at cluster C_q.

Step2: In the log record A_p^q if the ID of the accessing cluster is the same as the ID of cluster C_q, that indicates local access is made ($C_q = C_j$), then nothing need to be done.

Step3: In the log record A_p^q if the ID of the accessing cluster is different than ID of cluster C_q, that indicates remote access is made ($C_q \neq C_j$), then proceed to the next step.

Step4: Find out the total number of accesses made to the fragment F_i from each accessing remote cluster(s).

Letn_i^crepresents the total number of accesses made to the fragment F_i allocated at cluster C_q by each cluster C, where C = 1,2,3,...,Z. If (n_i^c<=α) then nothing need to be done, otherwise proceed to the next step.

Step5: Find out the average volume of data transmitted between fragment F_i and all clusters (including cluster C_q where fragment F_i is stored) from where accesses to the fragment F_i are made ($V_i^c t$) and also find out the average volume of data transmitted between fragment F_i and each remote cluster(s) C_j from where accesses to the fragment F_i are made ($V_i^j t$) where C = 1,2,3,....,Z and $C_j \neq C_q$, then proceed to next step.

The average volume of transmitted data can be determined using equation – (1). LetA_p^q V_i^c represents the volume of data transmitted between the fragment F_i allocated at cluster C_q and the cluster C in the access_logA_p^q, where C = 1,2,3,...,Z. Now consider$V_i^c t$ represents the average volume of data transmitted between the fragment F_i allocated at cluster C_q and the all accessing cluster C, then:

$$V_i^c t = (\sum A_p^q V_i^c) / \sum n_i^c \qquad (1)$$

Step6: If every accessing remote cluster(s) $C_j$fails to qualify the condition ($V_i^j t > V_i^c t$), then nothing need to be done, otherwise proceed to next step.

Step7: If the condition ($V_i^j t > V_i^c t$) is TRUE for only one remote accessing cluster C_j, then the fragment $F_i$is reallocated to cluster C_j and removed from the current cluster C_q and the site fragment allocation matrix at each cluster isupdated accordingly, otherwise proceed to next step.

Step8: If the condition ($V_i^j t > V_i^c t$) is TRUE for more than one remote accessing clusters $C_j$simultaneously, then find out the distance between the cluster C_q where fragment F_i is allocated and the clusters which qualified the condition. The fragment $F_i$is reallocated to the cluster which is at maximum distance from the cluster C_q and removed from the current

cluster C_q and the site fragment allocation matrix at each cluster is updated accordingly.
By reallocating fragment F_i to cluster which is at maximum distance from current cluster $C_q$delay in transactions can be reduced, which results improved system performance.

5. ILLUSTRATIVE EXAMPLE

To demonstrate how the proposed CADDA algorithm works, consider a distributed database system with three clusters and eight sites as shown in fig.2. These clusters are located at some distance (in km.) from one another as shown in the following table 4:

Table 4. Cluster Distance Matrix

Cluster	C_1	C_2	C_3
C_1	0	700	900
C_2	700	0	600
C_3	900	600	0

Only the respective row of cluster distance matrix is stored at each cluster. Consider that global relations in DDS have total twelve fragments (F_1, F_2, F_3, ..., F_{12}). Each fragment is allocated at one or more sites as per scheme shown in table 5:

Table 5. Site Fragment Allocation Matrix

Fragment→ Site↓	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
S_1	1	1	0	0	0	0	0	0	0	0	0	0
S_2	0	0	0	0	0	0	0	0	1	1	0	0
S_3	0	0	1	1	0	0	0	0	0	0	0	0
S_4	0	0	0	0	1	0	0	0	0	0	1	0
S_5	0	0	0	0	1	0	0	0	0	0	0	1
S_6	0	0	0	0	0	1	1	0	0	0	0	0
S_7	0	0	0	0	0	0	0	1	1	0	0	0
S_8	1	0	0	0	0	1	0	0	0	0	0	0

$$SFAM(S_j, F_i) = \begin{cases} 1, \text{ fragment } F_i \text{ is allocated to site } S_j \\ 0, \text{ otherwise} \end{cases}$$

Table 6. Access_Log at cluster C_1

AFID	ACID	ADateTime	DataVol
F_1	C_1	21-Feb-2017 12:26:15	180
F_2	C_1	21-Feb-2017 16:30:50	159
F_2	C_2	22-Feb-2017 11:55:57	270
F_2	C_3	22-Feb-2017 14:12:26	260
F_2	C_3	23-Feb-2017 10:44:33	300
F_2	C_2	23-Feb-2017 15:16:17	274
F_2	C_2	24-Feb-2017 12:18:24	356
F_2	C_3	24-Feb-2017 14:28:42	250
F_1	C_1	24-Feb-2017 16:32:48	245
F_2	C_2	25-Feb-2017 15:20:30	- - -
F_2	C_3	25-Feb-2017 15:20:30	- - -

At a particular moment the records of Access_Log table at cluster C_1 is shown in table 6. The Access_Log at cluster C_1 displays that total number of accesses made from clusters C_2 and $C_3$for fragment F_2 till 24-Feb-2017, is 3. i.e. $n_2^2 = n_2^3 = \alpha = 3$. On 25-Feb-2017 due to simultaneous read access, the total number of accesses made from clusters C_2 and $C_3$for fragment F_2 becomes 4, i.e. $n_2^2 = n_2^3 = 4 > \alpha$. Since there may be possibility of reallocation of fragment F_2, the last two transactions are kept on hold till the determination of fragment reallocation decision.
The following data transfer between different clusters for fragment F_2 can be calculated using Eq. (1):

- For fragment F_2,the average volume of data transferred between cluster C_1 and all other clusters = (159 + 270 + 260 + 300 + 274 + 356 + 250) bytes/7 = 1869 bytes/ 7 = 267 bytes
- For fragment F_2, the average volume of data transferred between cluster C_1 and cluster C_2 = (270 + 274 + 356) bytes/3 = 900 bytes/3 = 300 bytes
- For fragment F_2, the average volume of data transferred between cluster C_1 and cluster C_3 = (260 + 300 + 250) bytes/3 = 810 bytes/3 = 270 bytes

With above calculations, it is clear that 300 bytes >267 bytes and 270 bytes > 267 bytes, and $n_2^2 = n_2^3 = 4 > \alpha$ therefore both clusters C_2 and $C_3$are eligible for fragment F_2 reallocation. The cluster which is at more distance from cluster C_1 will be finally eligible for reallocation of fragment F_2. The distance between the clusters can be determined using Cluster Distance Matrix (CDM). As per CDM -

Distance between cluster C_1 and C_2: D_1^2 =700 km and
Distance between cluster C_1 and C_3: D_1^3 =900 km

So fragment F_2 is reallocated to cluster C_3 and deleted from the cluster C_1 and accordingly site fragment allocation matrix is modified at every cluster. As compare to cluster C_2, cluster C_3 is at more distance from current cluster C_1 where fragment F_2 is allocated, has to travel a lot, which results delay in operation. Reallocation of fragment F_2 to cluster C_3 results faster access. Similarly,for non-redundant distributed database system dynamic reallocation of data fragments can be performed.

6. COMPARISON

The comparison of proposed CADDA algorithm for dynamic data allocation in redundant and non-redundant distributed database system is performed with other cluster based algorithms introduced in [4,5.6] as well as with other non-cluster algorithms introduced in [7,8,9] on the basis of various properties as presented in table 7 and table 8:

Table 7. Comparison with cluster based algorithms

Paper Reference → Property↓	CADDA Algorithm	[4,5,6]
Cluster to cluster dynamic data reallocation capability	Yes	No
No. of remote data accesses	Less	More
Network overhead	Less	More
Efficiency	Higher	Lower

In [4,5.6] clusters are created first, thereafter data fragments are allocated to them. This data allocation is static.Therefore, initial allocation may not be appropriate after some time due to changing data access patterns and caused more number of remote data accesses. This will lead to more traffic on the network and take more time to respond a query. While in CADDA algorithm data fragments are allocated dynamically with respect to the changing data access patterns, which results less remote data accesses, less network traffic and consequently take less time to respond a query. In proposed approach, since data fragments are reallocated dynamically as per changing data access patterns of the clusters, complexity is slightly more as compared to static data allocation approach, but the system performance is better.

Table 8. Comparison with non-cluster based algorithms

Paper Reference → Property↓	CADDA Algorithm	[7,8,9]
Network sites clustering capability	Yes	No
No. of communications and communication cost	Less	More
Network overhead	Less	More
Efficiency	Higher	Lower

The CADDA algorithm facilitates clustering of network sites while algorithms proposed in [7,8,9]do not support clustering. Due to clustering of sites in CADDA algorithm, number of communications is less that results less communication cost as compared to algorithm proposed in[7,8,9]. Less number of communications in CADDA algorithm caused less network traffic, which consequently take less time to respond a query. Therefore, efficiency of CADDA algorithm is higher than algorithms proposed in [7,8,9].
Comparison results shown in table 7 and table 8 provedthat CADDA algorithm is more efficient and capablethan all other algorithms introduced in [4,5,6,7,8,9] and enhances the performance of the system.

7. CONCLUSION

In distributed database system, data allocation is a prominent performance aspect. System performancehighly depends on the efficient allocation of data fragments to sites.This paper discussed the performance improvement achieved by the proposed clustering method and CADDA algorithm. This clustering method groups the distributed database system network sites into clusters, and decrease the communication overhead between the network sites. CADDA algorithm further reduced network overhead by decreasing remote data accesses by dynamically reallocating data fragments to clusters as per changing data access patternsin redundant and non-redundant distributed database system. The comparative evaluation with other cluster and non-cluster based algorithms showed that proposed approach significantly improves the performance of distributed database system. This conclusion further needs more experiments and investigations. Therefore, as future work investigation of the proposed approach can be performed on larger scale networks consisting of more number of database sites. Moreover, for further improving the system performance,different clustering techniques, strong criteria for grouping database sites, and soft computing techniques can be explored.

8. REFERENCES

[1] M.T. Ozsu, P.Valduriez,"Principles of Distributed Database Systems," 3rd edn. Springer Science+Business Media, LLC 2011: New York, USA, 2011.
[2] S. Ceri, G.Pelagatti, "Distributed Databases: Principles &Systems," edn. 2008. McGraw-Hill International: New Delhi, India, 2008.
[3] R.Kumar, "Distributed database security: issues and challenges," Proceedings of Int. Conf. Computing: Updates & Trends, Ghaziabad, India, 2010, 301-309.
[4] I. Hababeh, M. Ramachandran, N. Bowring,"A high-performance computing method for data allocation in distributed database systems," J Supercomput. 2007; 39(1): 3–18.
[5] I.Hababeh, "Improving network systems performance by clustering distributed database sites,"The Journal of Supercomputing, 2012; 59: 249-267.
[6] R.M.H Al-Sayyed, F.A Al-Zaghoul,D. Suleiman, M. Itriq, I. Hababeh,"A new approach for database fragmentation and allocation to improve the distributed database management system performance," Journal of Software Engineering and Applications. 2014; 7: 891-905.
[7] N. Mukharjee,"Non-replicated dynamic fragment allocation in distributed database systems," Springer-Verlag Berlin Heidelberg, CCSIT, Part I, CCIS 131, Bangalore, Jan. 2011, 560-569.
[8] R. Kumar, N. Gupta, "An extended approach to non-replicated dynamic fragment allocation in distributed database systems," IEEE Xplore, ICICT, Ghaziabad, India, 2014, 861-865.
[9] R. Kumar, N. Gupta,"An extended efficient approach to dynamic fragment allocation in distributed database systems," International Journal of Control Theory and Applications (IJCTA). 2016; 9(20): 473-482.
[10] T. Ulus, M. Uysal,"Heuristic approach to dynamic data allocation in distributed database systems," Pakistan Journal of Information and Technology. 2003; 2(3): 231-239.
[11] A. Singh, K.S Kahlon,"Non-replicated dynamic data allocation in distributed database systems," International Journal of Computer Science and Network Security, 2009; 9(9): 176-180.
[12] R. Kumar, N. Gupta, "Non-redundant dynamic data allocation in distributed database systems," International Journal of Computer Applications, 2012; Special Issue on ICNICT, 6: 06-10.

[13] S. Kamali, P. Ghodsnia, K. Daudjee, "Dynamic data allocation with replication in distributed systems," IEEE Explore, 2011, 978-1-4673-0012-4/11/$26.00.
[14] H.I Abdalla,"A new data re-allocation model for distributed database systems,"International Journal of Database Theory and Application. 2012; 5(2): 45-59.
[15] A.A Amer, H.I Abdalla,"A heuristic approach to re-allocate data fragments in DDBSs," Information Technology and e-Services (ICITeS) Int. Conf., Sousse, Tunisia, March 2012, 01-06.
[16] R. Kumar, N. Gupta,"Dynamic data allocation in distributed database systems: a systematic survey," International Review on Computers and Softwares, 2013; 8(2): 660-667.
[17] Y. Zhu,J. Liu, M. Guo, W. Ma, G. Yi, Y. Bao, "ACIA, not ACID: conditions, properties and challenges," arXiv:1701.07512v2 [cs.DC], 2017 January; 1-8.