International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)
Volume VIII, Issue IV, April 2019 | ISSN 2278-2540

Balancing Fragment by Using Group Based Approach for Partial-Replicated Dynamic Fragment Allocation in Distributed Database System

Nang Khine Zar Lwin[1], Tin Myint Naing [2]
[1,2]*Faculty of Information & Communication Technology*
University of Technology (Yadanarpon Cyber City), Myanmar

***Abstract*—Distributed database system is very useful for controlling data which is distributed over different machines on multiple locations. One of the design issues of distributed databases is how to make decisions about the placement of data and programs across the sites of a computer network as well as possibly designing the network itself. The best way of data allocation is to place the data fragments to the sites from which they are most frequently accessed to reduce the total data transfer cost while executing a set of queries in the realistic dynamic environment, where the access pattern of fragments from different sites change over time. In this paper, a group-based strategy for partial-replicated dynamic fragments allocation in distributed database system is proposed. The proposed algorithm has been used to dynamically reallocate the fragments by considering group-based in partial-replicated distributed database system. The proposed approach is implemented on a sample distributed database system provides better response time and will improve the overall of distributed database system performance**.

***Keywords*—Distributed Database; Fragmentation; Static Fragment Allocation; Dynamic Fragment Allocation; Group of sites.**

I. INTRODUCTION

Distributed database technology is one of the most important developments of the past two decades in the field of database systems. Due to demand for the emergences of new technology, quick and reliable flow of data transfer rate and enable by advances database theory, communicate technology and distributed database systems. Distributed database is considered as data that are stored to different machines which are connected to some network, via internet, intranet or extranet that looks like one centralized database to the end user [2]. Distributed database systems are used in applications requiring access to an integrated database from geographically dispersed locations. As a major objective of distributed databases, data fragmentation and allocation are two of the critical aspects to provide ease of access to data for users at many different locations. Distributed database management system governs the storage and processing of logically related data over interconnected computer system in which both data and processing functions are distributed among several sites [3].

The case for the improved performance of distributed databases is to design the optimal fragmentation, replication and allocation of the underlying databases. In optimal fragmentation, replication and allocation could be very complex. The best strategy of allocating fragment is the case to improve performance, to increase the availability, and access facility of data. The main objective of fragment allocation is enabling data to be stored in close proximity to its points of use. There are many heuristic algorithms are proposed for allocation of data or fragment in distributed database system. Each fragment can be stored to one or more sites among the connected network to access data availability, to improve system reliability includes reduce complexity, reliability distribution and redundant processing.

In recent years, there are many fragment allocation approaches has been proposed for static environment to design a database depend on some static access patterns and dynamic environment where data access patterns can change over time. In this paper, it focuses on approach of dynamic fragment allocation algorithm with partial-replication in distributed database systems. In the partitioned scheme the database is divided into a number of disjoint partitions each of which is placed at a different site. Replicated designs can be either fully replicated (also called fully duplicated) where the entire database is stored at each site, or partially replicated (or partially duplicated) where each partition of the database is stored at more than one site, but not at all the sites [2]. This is based on the belief that the frequency of accessing database tables is not uniform. In this paper a group-based approach to dynamic fragment allocation algorithm for partial-replicated is proposed which dynamically reallocates fragment among different sites by considering in group-based. In reallocation process, this algorithm can handle the case when more than one sites qualify to the fragment in partial-replicated distributed database system and can give better response for the queries of the sites that access frequently according to consider in group.

The organization of the rest of this paper is as follows. Section II describes the related work of the system. Section III discuses the group-based distributed database environment. Section IV provides the proposed group-based strategy for partial-replicated dynamic fragment allocation in distributed database system. Section V explains the result of implementation of the system model. Section VI comparison and Section VII summarizes the contribution of the study.

II. RELATED WORK

Many studies have been done several data allocation approaches based on static data allocation strategy over past few years. The task of static data allocation over the different site of a computer network in distributed database is to the design of a database depending on some static data access patterns or static query patterns. Moreover, several works have been published dynamic data allocation algorithm in database system. A model for dynamic data allocation for data redistribution was introduced by [4]. In [5] machine learning was adapted to an approach for allocating fragments. There are many heuristic approaches have been explained by [6]. Dynamic data allocation algorithm is proposed by [7] which reallocate data with respect to changing data access pattern. In [8] 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 [15] an algorithm called Threshold and Time Constraint Algorithm (TTCA), the fragment reallocates according to the changing data access patterns with time constraint. The problem of this algorithm can cause scaling problem. To remove the scaling problem of TTCA, an Extended Threshold Algorithm (ETA) was proposed in [9]. I Extended Threshold Algorithm solved the scaling problem and decreased the space requirement as time constraint is not stored. In [10] Threshold Time Volume and Distance Constraints which additionally include volume of data transmission besides access threshold, time constraints of database access was proposed. [1] proposed extended dynamic fragment allocation algorithm which is the extension of [10]. This algorithm solved the problem [7,8,19,10] that it is more than one site quality for the fragment relocation by select the site which occupy the fragment. To allocate data dynamically in redundant distributed database system, [11] introduced an algorithm which was based on fragment's correlation, lazy replication strategy, and non-uniform distances between network sites.

Many clustering method for data allocation were proposed to minimize number of communications and associated communication cost. In [13] NNA algorithm for dynamic data allocation methods in distributed database system was presented. In NNA algorithm, the fragment is migrated to the nearest neighbor of the source node and placed in the path to the node which has the maximum access counter. This algorithm is suitable for low bandwidth and the large fragment size. In [12] RFA algorithm which considers the highest frequency of access to a fragment by region instead of individual node was proposed. This algorithm migrate the fragment to the site with max count in the region which has maximum count of accessing particular fragment than other regions. In [14] 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 [16] by combining three enhanced techniques - database fragmentation, network sites clustering and data allocation. In this paper, a group-based approach to partial-replicated dynamic fragment allocation algorithm is proposed. The proposed algorithm intends to take account user access patterns and reallocate the related fragments dynamically to give better response for the queries by considering in group factor where the access time of sites made access to fragment is greater than access threshold value within time constraint interval for fragment reallocation. It will also reallocates fragment by selecting the maximum update data request of accessing fragment to give better response for the queries while there are more than one accessing site.

III. GROUP BASED DISTRIBUTED DATABASE ENVIRONMENT

This section discusses the manner in group based distributed database environment and duplicates data for storage. Distributed databases is defined as databases located at different machines at the same or different locations that looks like one centralized database to the end user[2]. Assume that there is a fully connected network consisting of four sites in distributed database system placed at some distance respectively as following Table 1 N= {N_1, N_2, N_3, N_4} and four fragment or partitions P={p_1,p_2,p_3,p_4} of global relations in DDS. The distance shown in Table I is in km. Then, sites N1, N2 are setup into same group and respectively sites N_3, N_4 in another group. All partitions or fragments are allocated in partial-replicated manner on to the four sites in distributed database system like as Fig 1.

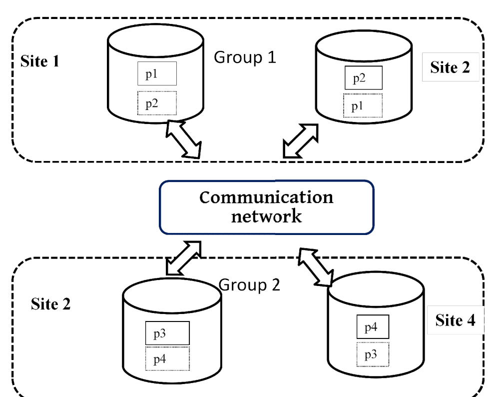

Fig. 1. Distributed Database Environment

TABLE I
Site Distance Matrix

Site	N1	N2	N3	N4
N1	0	100	300	450
N2	100	0	200	350
N3	300	200	0	150
N4	450	350	150	0

Each site stores one or more replica that is, copies of the fragments or partitions assigned to the site group of which the site is a member. Each site should be located on a separate computer. A site group consists of one or more sites and stores fragments, or sets of replicas. Fragment is a partition of the data stored by group. Each site is responsible for keeping at least one copy of any fragments assigned to it. Each site in a node group stores a replica. A replica belongs entirely to a single site and a site store several replicas. Fragment 1 or p1 is stored on site group 1; a primary fragment is stored on site 1, and a backup copy of the fragment is stored on site 2. Fragment 3 or p3 is stored on the other site group 2; this fragment's primary is on site 3 and its backup replica is on site4. Fragment 2 or p2 is stored on site group 1. However, the placing of its two replicas is reversed from that of fragment 1 or p1; for fragment 2 or p2, the primary replica is stored on site 2, and the backup on site 1. Fragment 3 or p3 is stored at the same way of fragment 2 or p2 in group 2.

IV. PROPOSED GROUP BASED PARTIAL-REPLICATED DYNAMIC FRAGMENT ALLOCATION ALGORITHM

In this section, the overall procedure of the proposed algorithm is shown in Fig. 2. The fragments are assigned to distributed database system sites in partial-replication manner. Each fragment may be allocated at one site and the copy of fragment is allocated in another site in same group. Two sites are assigned into one group. Each site stores pre-specified access threshold value and a pre-specified time interval which is used to control the re-allocation process. Then, access log information (identifier of the fragment accessed, address of accessing site, date and time of accessed, read or write of accessed, volume of data transmitted to or from accessed fragment in bytes) which include some information for each fragment access to that site assigned in each site. Moreover, assume that there is a specific distance between two sites in kilometer. There are three main parts in the proposed algorithm; obtaining and analyzing query Access log, checking priority with evaluated rules before migration, balancing fragments group after migration. The used notations in this paper are described in Table II.

The proposed approach which additionally modified read and write data volume factor to Threshold Time Volume and Distance Constraints Algorithm [1]. The system intends to take account user access patterns and reallocate the related fragments dynamically to give better response for the queries by considering in group. In reallocation process, the proposed algorithm reallocates the fragment by considering a concept of group instead of distance factors. The priority of group and node tables suggests making the correct decision in reallocation process.

TABLE II
Algorithm Notations

Notation	Meaning
F	Number of fragments or partitions of global relations in distributed database system
S	Number of sites in distributed database system
P_i	The i^{th} fragment
N_x	The x^{th} site
α	Access threshold for fragment relocation
β	Time constraint for fragment relocation
$A_z{}^y$	Log Information record z^{th} access at site y
n_i^x	Total number of read/write accesses from site N_x to the fragment P_i within time interval β up to current access time t
Vgj_i^x	Volume of read and write data transmitted between fragment Pi and site N_x in group gj within time interval β up to current access time t
D_x^y	Distance between site S_x and site S_y
Vw_i^x	Volume of write data between fragment P_i and site S_x within time interval β up to current access time t
Np	Number of fragments or partitions in each group
G	Number of groups in distributed database system

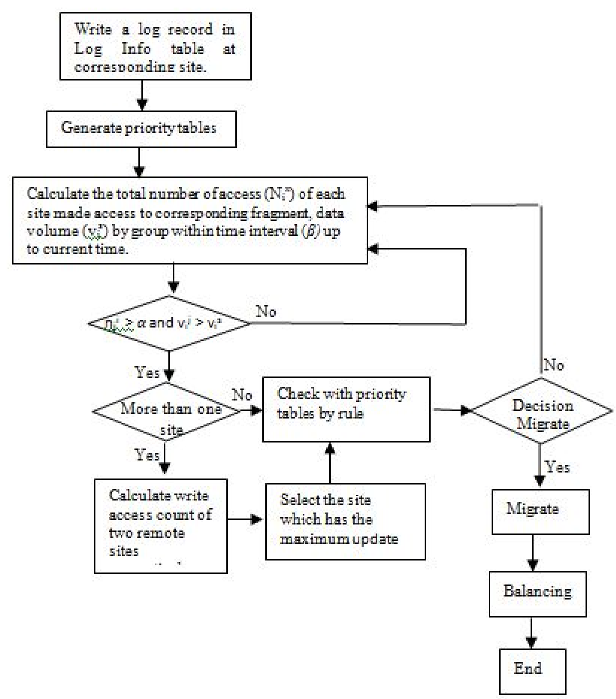

Fig. 2. The overall procedure of proposed algorithm.

A. Obtaining and Analyzing Query Access log

In this process, access log information at each site s_y is written every time when each fragment at owner site is accessed from different sites s_x where y= 1,2,3,... ,S , x= 1,2,3,... ,S, and y = x or x≠ y. Let site s_x can access fragment p_i allocated at site s_y at time t, where i= 1,2,3,..., F. Queries can be invoked by remote different sites (s_x) or local site (s_y). Two sites are defined into one group, with each group having approximately equal number of sites subject to the constraint that the sites are in near proximity to each other. Number of partitions or fragments in each group equal to Np= F/G. All the small fragments of global relation are initially allocated over different sites using any static allocation method in partial-redundant manner. One copy of each fragment is stored in the site that has the same group of the site that contains the primary fragment.

Each site store Log_Info table with structure of this table is Log_Info (AFID, ASID, ADateTime, RorWA, DataVol) where AFID means ID of the fragment which is accessed, ASID means ID of the site which accesses the fragment, ADateTime means date and time of fragment access from respectively accessing site, RorWA means read or write of fragment access and DataVol means volume of read data transmitted to and from the accessed fragment or volume of updated data. Each site stores an own access log record for each access to the fragments allocated to that site. Each Log_Info record is denoted by A_z^y (means z^{th} access at site y, where z=1,2,3, ... ∞ and y = 1,2,3, ... N). Query traces that are invoked from many clients via some applications are simplified to obtain read or write operation. The following steps are performed within time interval β up to current access time t.

Step 1: Write a log record A_z^y in Log_Info table at site s_y.

Step 2: Generate node priority table and group priority table by access count at each site make access to eight partitions.

Step 3: Calculate the total number of read and write accesses between the fragment p_i and each remote site s_x that made access to the fragment respectively where x=1,2,3,..S. If $f_i(N_j^x < \alpha$), then do nothing, otherwise to the following step.

Step 4: Calculate the average volume of read and write data transmitted between fragment p_i and all sites (including local site s_y) in each group that made access to the fragment p_i. Consider A_z^y Vgj_i^x denotes the volume of read and write data transmitted between the fragment f_i allocated at site s_y and all sites in each group that made access to the fragment p_i in the access A_z^y log record. Furthermore let $Vgj_i^x t$ denotes the average volume of read and write data transmitted between the fragment p_i allocated at site s_y and all sites in each group that made access to the fragment p_i occurred within time interval β up to current access time t then:

$$Vgj_i^x\ t\ = (\sum A_z^y\ V_a^x + \sum\ A_z^y\ V_b^x) \tag{1}$$

Step 5: If the volume of read and write data transmitted of each accessing remote group does not greater than the average volume of read and write data transmitted of accessing local group, then do nothing, otherwise go to the following step.

Step 6: If there is only one accessing remote site s_x qualify constraints stated in step 5, then check node priority and group priority with priority tables by evaluated rules.

Step 7: If more than one accessing remote sites qualify constraints stated in step 5. Calculate the volume of write data transmitted between the fragment p_i and all qualified remote sites within time interval β up to current access time t. The volume of write data can be calculated using equation – (2)

$$Vw_i^x t\ = (\sum A_z^y\ Vw_i^x\ ) \tag{2}$$

Then, select the site which has the maximum write data volume than other sites and, then check node priority and group priority with priority tables by evaluated rules.

B. Checking Priority with Evaluated Rules

In these parts, before migration process is performed, the priority tables are used to support to make the correct decision for fragment migration.

Step 8: If fragments or partitions need to migrate to farthest sites. Rule for farthest sites is shown in the following:

1. Number of partitions can be allocated in farthest sites<= F/Np
2. Suggest migrating or not

   (2.1) if (site priority >= priority1 and site priority <= priority (F/Np) and group priority < priority (Np+2))

   Suggestion =need to migrate

   (2.2) if (site priority >= priority(F/Np+1) and site priority <= priority (Np) and group priority < priority (Np+1))

   (2.2.1) check copy site

   if(copy site priority < =priority (Np) & group site < priority (Np+1))

   Suggestion =need to migrate copy site

   else if (copy site priority < =priority (Np) & group site > priority (Np+1))

   Suggestion= not to migrate

   else

   Suggestion= not to migrate

Step 9: If fragments or partitions wants to migrate to middle sites. Rule for middle sites is shown in the following:

1. Number of partitions can be allocated in middle sites <=(F/Np+1)

2. Suggest migrating or not

(2.1) if (site priority >= priority1 and site priority <= priority (F/Np+1) and group priority < priority (Np+2))

Suggestion =need to migrate

(2.2) if (site priority == priority(Np) &

group priority < priority (Np+1))

Suggestion =need to migrate

else if (site priority < =priority (Np) & group site > priority (Np+1))

Suggestion = not to migrate

else

Suggestion = not to migrate

Step 10: Reallocate the fragment to the suggested site and remove from the current site and then, update the fragment allocation information.

C. Balancing Fragments Group after Migration

After migration constraint stated with above steps, fragment sites are balancing by group-based in order to attend each group having approximately equal number of fragments for all sites in the group.

Step 11: Check which group has more partitions or fragments than constraint number of fragments in each group. If each group has equal fragments, then do nothing, otherwise to the following step.

Step 12: Find which fragments in group that has more fragments and migrate to group which needs partitions by checking with group priority table. . If there is no fragments need to migrate, and then do nothing, otherwise to the following step.

Step 13: Select the site which has the maximum update than another site in group and reallocate the fragment to that site and update the fragment allocation information.

V. IMPLEMENTATION RESULT

In the experimental result, Dell PC of core i5 processor with 4 GB RAM and 1 GB graphic card is used. Each site generates a set of queries which has 75 percent of retrieval queries and 25 percent of write queries. Access threshold value for fragment reallocation is set to 13 percent of the number of queries and 1 week for Time constraint for fragment reallocation. In this work, a global relation is partitioned into 8 subsets of rows by use of the selection operator. Suppose that there are eight fragments P={p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8} of global relations in DDS. Let there are four sites in distributed database system placed at some distance respectively in table I.

Then, sites N_1, N_2 are created into same group and respectively sites N_3, N_4 in group 2. All fragments of global relation are initially allocated to the four sites in distributed database system like as Fig. 3. These fragments are allocated in partial-replicated manner on different four sites as shown in Table III. After distributing all fragments over the different sites, each site stores pre-specified access threshold value α and a pre-specified time interval β and Log_Info Table. After migration process from every four sites is finished, the fragment allocation information is update and the fragment is reallocated in Fig. 4.

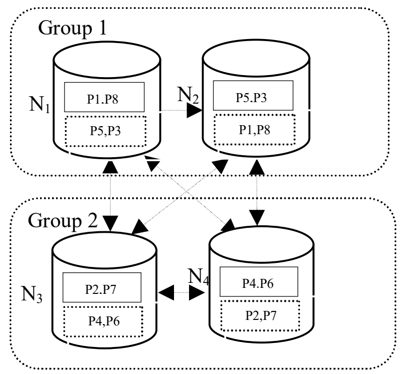

Fig.3. Fragment initially allocated over different sites

TABLE III
Fragment Allocation Information

Fragment	Primary Site	Copy Site
P_1	N_1	N_2
P_2	N_3	N_4
P_3	N_2	N_1
P_4	N_4	N_3
P_5	N_2	N_1
P_6	N_4	N_3
P_7	N_3	N_4
p_8	N_1	N_2

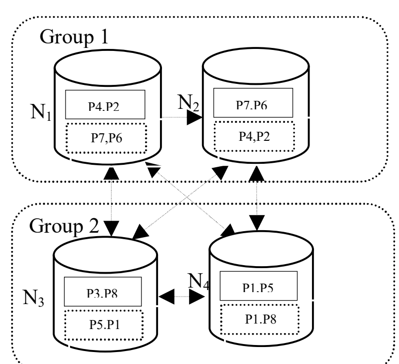

Fig. 4. Fragment reallocated over different sites

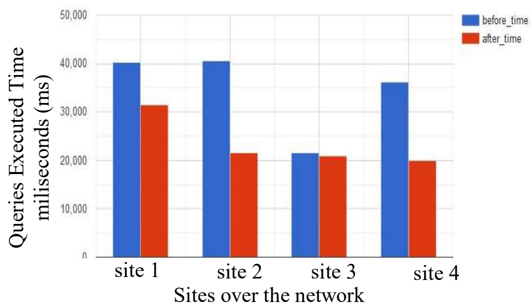

Fig.5 Executed Time of queries for four sites after and before applied proposed algorithm

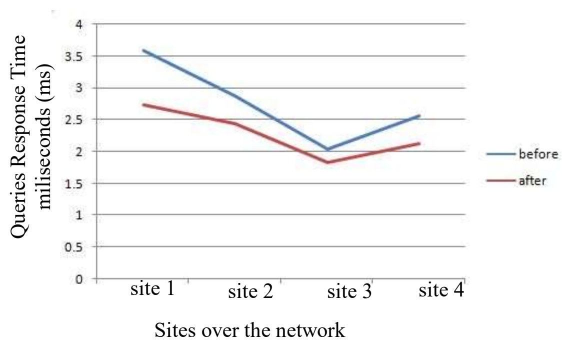

Fig. 6. Response Time of queries for four sites after and before applied proposed algorithm.

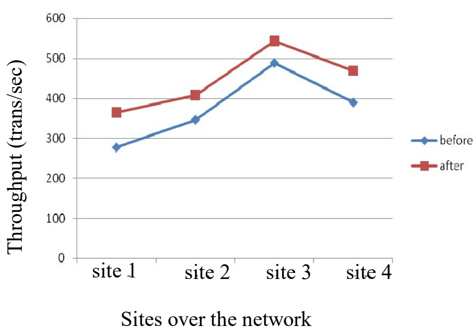

Fig.7 Comparisons Throughput of four sites after and before applied proposed algorithm

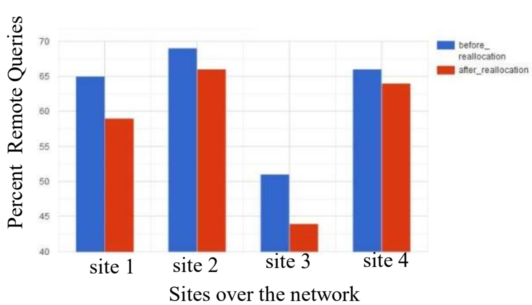

Fig.8 Percent of Remote Queries for Four Sites after Applied Proposed Algorithm

According to the result shows in Fig. 5 that the different of queries executed time before and after migration of site 1 and site 3 is not so significant, but it can be seen clearly decrease in site 2 and site 4. The queries response time of four sites is slightly decreased after migration as shown in fig. 6. The throughputs (transaction/sec) is measured and stated in fig. 7. The throughput after migration is increase a bit than the throughput before migration. After applied the proposed algorithm, the percent of remote queries for each site decrease respectively in fig. 8.

VI. Comparison

In this section, the main comparison indicators are the performance of un-balancing state and balancing state after migration. The test results are shown in Fig.9 and Fig.10.

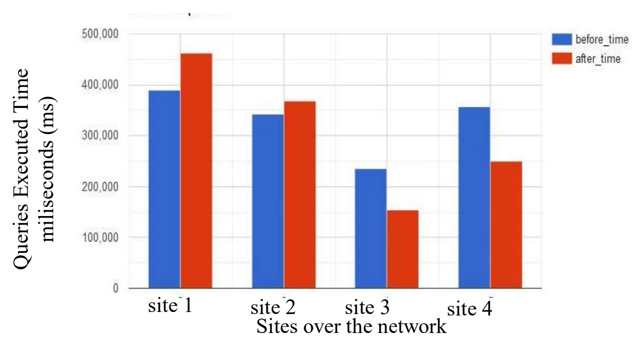

Fig. 9 Queries executed Time of four sites for un-balancing state after migration

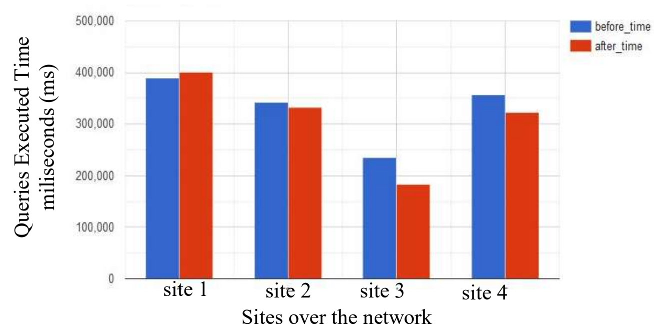

Fig. 10 Queries executed Time of four sites for balancing state after migration

According to the results of the system, the executed time of four sites are overall decreased after migration by balancing fragments in group. By using priority tables to suggest the correct decision before migration process that can reduce total access time of four sites than un-balancing fragments in group after migration.

VII. Conclusion

In conclusion, distributed databases are being increasingly used in various organizations and the critical issue is how to distribute global database and how to control other affecting transfer cost and the overall system of performance. The decision on allocation of fragment over different site is important on performance of distributed database system. In this work, the group-based algorithm for partial-replicated dynamic fragments allocation in distributed database system is presented. TTVDCA algorithm can improve the response time of the furthest sites but sometimes the response time of other sites can decrease in this case. The proposed algorithm reallocates data with respect to the changing data access patterns by considering in group and balancing approach so that response time of all sites of each group is not so different each other. The proposed algorithm will be tolerant the problem of site failure that can affect some database and it also reduce total transfer cost from different sites and total access time.

ACKNOWLEDGMENT

The authors would like to thank the help they received from Faculty of Information and Communication Technology at University of Technology (Yadanarpon Cyber City) for supporting the requirements facilities to complete well this work.

REFERENCES

[1]. Raju Kumar, Neena Gupta , "An Extended Approach to Non-Replicated Dynamic Fragment Allocation in Distributed Database Systems", International Conference on Issues and Challenges in Intelligent Computing Techniques(ICICT), 2014 IEEE.

[2]. M.T. Ozsu, P.Valduriez,"Principles of Distributed Database Systems," 3rd edn. Springer Science+Business Media, LLC 2011: New York, USA, 2011.

[3]. Ali A. Amer, Hassan I, Abdalla ,"A Heuristic Approach to Re-Allocate data Fragments in DDBSs, Department of Computer Science, College of Computer & Information Sciences, King Saud University ,2012 IEEE

[4]. O. Wolfson, S. Jajodia, and Y. Huang, "An Adaptive Data Replication Algorithm," ACM Trans. Database Systems, vol. 22, no. 2, pp. 255-314, 1997.

[5]. Chaturvedi, A. Choubey, and J. Roan, "Scheduling the Allocation of Data Fragments in a Distributed Database Environment: A Machine Learning Approach," IEEE Trans. Eng. Management, vol. 41, no. 2, pp. 194-207, 1994.

[6]. Y. Huang and J. Chen, "Fragment Allocation in Distributed Database Design," J. Information Science and Eng., vol. 17, pp. 491- 506, 2001.

[7]. A. Brunstroml, S.T. Leutenegger and R. Simhal, "Experimental Evaluation of Dynamic Data Allocation Strategies in a Distributed Database with changing Workload", ACM Trans. Database Systems, 1995.

[8]. 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.

[9]. 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.

[10]. 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.

[11]. 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.

[12]. Tarun Gulyani , Pallath Paul Varghese, "Region based Fragment Allocation in Non-Replicated Distributed Database System", Interantional Journal on Advanced Computer Therory and Engineering (IJACTE) 2319-2526, Vol-1, Issue-1, 2012.

[13]. Basseda, R., Tasharofi, S., Rahgozar, M., Near Neighborhood Allocation (NNA): A Novel Dynamic Data Allocation Algorithm in DDB, In proceedings of 11th Computer Society of Iran Computer Conference (CSICC2006), Tehran, 2006.

[14]. I.Hababeh, "Improving network systems performance by clustering distributed database sites,"The Journal of Supercomputing, 2012; 59: 249-267.

[15]. Nilarun Mukharjee, "Non-Replicated Dynamic Fragment Allocation in Distributed Database Systems", CCSIT 20 II, Part I, CCIS 131, pp. 560- 569, Springer-Verlag Berlin Heidelberg, 2011.

[16]. S. Agrawal, V. Narasayya, and B. Yang, "Integrating Vertical and Horizontal Partitioning into Automated Physical Database Design," Proc. 2004 ACM SIGMOD International Conf. Management Data, pp.359-370.