International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

# Partial-Replicated Dynamic Fragment Allocation in Distributed Database System

**Nang Khine Zar Lwin[1], Dr. Tin Myint Naing[2]**

[1]Faculty of Information & Communication Technology, University of Technology (Yadanarpon Cyber City), Pyin Oo Lwin, Myanmar

**Abstract:** ***Many organizations having a number of databases may geographically distribute databases all over the world using distributed database system. One major task of distributed database is how to allocate fragments among the different sites over the network. It has the greater impact on the performance of the distributed database and the operational efficiency of the system. Nowadays, many application or invoked query from different sites where the access patterns of sites made to fragments change over time and then dynamic fragment allocation technique provides better solution than static fragment allocation technique. The dynamic fragment allocation technique migrate the fragment dynamically according to the access patterns of each fragment. This paper proposes a strategy for partial-replicated dynamic fragments allocation in distributed database system. In reallocation process, the proposed strategy reallocates the fragment with respect to read and write data volume factor. When more than one site simultaneously qualifies for fragment reallocation, it will reallocate fragment with respect to the write data volume and distance factors of respectively different sites. This algorithm will decrease the transfer cost and be faster response time of the system.***

**Keywords:** Distributed Database, Fragmentation, Static Fragment Allocation, Dynamic Fragment Allocation.

## 1. Introduction

Today's business environment has an increasing need for distributed database as the desire for consistent, scalable, reliable and accessible information is steadily growing. 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. It is actually a set of server machines working in synchronization to cater the needs to multiple users. 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. The different geographic of a computer network can communicate through a network and it is managed by a distributed database management system [5].

The primary issue of a distributed database system is to design the fragmentation, replication and allocation of the underlying databases. In optimal fragmentation, replication and allocation could be very complex. The task of allocating fragment in a distributed database system has a critical impact upon the reliability and performance of the system [2,6]. It is very important to decide which fragment is stored closer to the site where they are most frequently used. As a result, it is to get better performance for allocating fragments and then reduces the amount of irrelevant data accessed by the transactions of the database.

In recent years, many approaches have been proposed for allocation of data or fragment in distributed database system. In these approaches, fragment allocation has been proposed for static environment to design a database depend on some static access patterns. But, the static allocation approaches may not provide the best solution for a 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 partially replicated database, the designer will make copies of some of the tables (or fragments) in the database and store these copies at different sites [10]. This is based on the belief that the frequency of accessing database tables is not uniform. The proposed algorithm is based on TTCA by A. Singh, and K.S.Kahlon [3]. This approach already include access threshold, access time constraints. N.Mukharjee [4] additionally added the volume of data transmission. Raju Kumar, Neena Gupta [1] proposed an extended approach to non-replicated dynamic fragment allocation in distributed database systems. The proposed algorithms is additions of Threshold Time Volume and Distance Constraints Algorithm [1] by analyzing data read and writes operation instead of data volume only and reallocate the related fragments dynamically to closer site. The system intends to take account user access patterns (data read and write operation) and reallocate the related fragments dynamically to give better response for the queries.

The organization of the rest of this paper is as follows. Section II describes the related work of the system. Section III provides the proposed algorithm for partial-replicated dynamic fragment allocation in distributed database system. Section IV explains the result of implementation of the system model. Section V summarizes the contribution of the study.

## 2. Related Work

The task of allocating data or fragment over the different site of a computer network in distributed database is an important activity. Many studies have been published on the problem of fragment allocation to nodes. Many researchers have been done several data allocation approaches based on static data allocation strategy over past few years. Moreover, several works have been published dynamic data allocation algorithm in database system. In [16] a model for dynamic data allocation for data redistribution was introduced. The approach for allocating fragments by adapting a machine learning approach was provided by [19]. In [17] an approach based on lagrangian relaxation was provided and heuristic approaches have been explained by [13]. Moreover, a mathematical modeling approach and a genetic algorithm base approach to allocate operation to nodes have been presented by [8]. [9] proposed a high-performance computing method for data allocation in distributed database system.

Dynamic data allocation algorithm is proposed by [18] which reallocate data with respect to changing data access pattern. In [11] an optimal algorithm for non- replicated database system is proposed. This algorithm depends on the changing frequency of access pattern for each fragment. This algorithm can cause some problem that it will spend more time for transferring fragments if the changing frequency of access is high. Threshold algorithm for non- replicated distributed database was proposed by [19].This algorithm reallocated the fragments respect with the changing data access patterns. It depends highly on threshold value. In [3] Threshold and Time Constraint Algorithm (TTCA) was proposed which the fragment reallocates according to the changing data access patterns with time constraint. The problem of this algorithm can cause scaling problem. In [12] Extended Threshold Algorithm was developed which solved the scaling problem and decreased the space requirement as time constraint is not stored. In [4] Threshold Time Volume and Distance Constraints which additionally include volume of data transmission besides access threshold, time constraints of database access was proposed. It does not say how to solve the problem in [3,4,12,18] that it is more than one site quality for the fragment relocation by selecting the site which occupy the fragment. [1] proposed extended dynamic fragment allocation algorithm which is the extension of [4] by adding distance factor besides existing factors to solve the problem in [3,4,12,18].

In [6] a new framework for a dynamic fragment allocation that considers replication and fragment correlation under a flexible network topology is proposed. This algorithm relies on collecting data fragment access patterns. A data re-allocation model for replicated and non-replicated is proposed by [7]. In this algorithm, re-allocation process based on selecting the maximal updates cost value for each fragment and deciding on the re-allocation accordingly. In this paper, an approach to partial-replicated dynamic fragment allocation algorithm is proposed. In the proposed algorithm, re-allocation processes is performed according to read and write data volume made access to fragment from respectively different sites and distance factor where the access time of sites made access to fragment is greater than access threshold value within time constraint interval for fragment reallocation. Moreover, the problem of site failure which cause of some reasons can be tolerant because a copy of every each fragment is stored in other different site.

## 3. The Proposed Partial-Replicated Dynamic Fragment Allocation Algorithm

### 3.1 Problem Description and Notations

In this section, the overall procedure of the proposed algorithm is shown in Fig. 1. Assumed that there is a fully connected network consisting of many sites N= {$N_1$, $N_2$, $N_3$, ... ,S }, 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. Firstly, global relation is partitioned into small fragments or partitions and distributes the fragments in partial-replicated allocation over the different sites of a network using a static data allocation method. Each site contains at least one fragment. The overall procedure of algorithm is divided into three phases; fragmentation phase, allocation phase and migration phase. The used notations in this paper are described in Table I.

The proposed algorithm is based on (TTVDCA) [1] algorithm and it additionally considers update data volume besides distance factors in reallocation fragment process. TTVDCA algorithm selects the site which is at maximum distance from the current site to reallocate fragment when there are more than one site simultaneously qualify for fragment reallocation. In reallocated fragment process, the proposed algorithm selects the site which has maximum update data volume access to fragment and also selects the maximum distance from the current site when the update data volume of the two qualify sites is not significantly different.

**Table 1**: 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 |
| $nr_i^x$ | Total number of read accesses from site $N_x$ to the fragment $P_i$ within time interval β up to current access time t |
| $Vr_i^x$ | Volume of read data transmitted between fragment $P_i$ and site $N_x$ 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 |

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

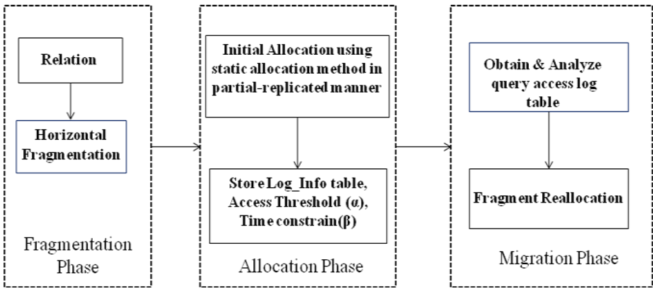


**Figure 1:** The overall procedure of proposed algorithm.

### 3.2 Fragmentation and Replication

Fragmentation approach breaks a table up into two or more pieces called fragments or partitions and allows storage of these pieces in different sites. Each fragment can be stored at any site over a computer network. Information about the fragmentation is stored in the distributed data catalog from which it is accessed by the transaction processor to process user requests. Data fragmentation strategies are based at the table level and consist of dividing a table into logical fragments.

Horizontal fragmentation refers to the division of a relation R into subsets of rows by use of the selection operator.

$$R^j = \sigma_{Qj}(R), \text{ where } 1 \leq j \leq m \quad (1)$$

Where $Q_j$ is the selection conditions as a simple predicate and m is the maximum number of fragments. Similar to vertical fragmentation, horizontal fragmentation must be done in such a way that the base table can be reconstructed. Because each fragment contains a subset of the rows in the table, Horizontal fragmentation can be used to enforce security and/or privacy of data. Every horizontal fragment must have all columns of the original base table.

Data replication refers to the storage of data copies at multiple sites served by a computer network. Fragmented copies can be stored at several sites to serve specific information requirements. Because the existence of fragmentation copies can enhance data availability and response time, data copies can help to reduce communication and total query costs. Replicated data is subjected to the mutual consistency rule. The mutual consistency rule requires that all copies of data fragments be identical. Three replication scenarios exist:

1) A fully replicated database: Stores multiple copies of each database fragment at multiple sites. It can be impractical due to the amount of overhead it imposes.
2) A partially replicated database: Stores multiple copies of some database fragments at multiple sites. It is handled well by the most databases.
3) A non-replicated database: Stores each database fragment at single sites.

### 3.3 Allocation Phase

Some requirement activities and initial allocation process are performed in this phase. Suppose that there are F fragments of global relations and N sites in the distributed database system. Two parameters- Access Threshold for fragment reallocation (α) and Time constraint for fragment reallocation (β) are stored in each site over the network before accessing fragment.

Step 1: Define two sites 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.
Step 2: 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.

Step 3: Store Log_Info table at each site. The 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)

### 3.4 Migration Phase

In this phase, Log_Info at each site $S_x$ is written every time each fragment is accessed from different sites $S_y$ where y= 1,2,3, ... ,N, x= 1,2,3, ... ,N, and y = x or x≠ y. Let site $N_x$ can access fragment $P_i$ allocated at site $N_y$ at time t, where i= 1,2,3, ... F. Queries can be invoked by remote different site ($N_x$) or local site ($N_y$) to access a fragment allocated to the local site $N_y$ where the local agent performs the following steps 1 to 4. Query traces that are coming from the clients via some application are analyzed to get read or write operation. Read and write sets of transactions are computed to access to respectively data volume.

Step 1: Write a log record $A_z^y$ in Log_Info table at site $N_y$.
Step 2: If the ID of local site $N_x$ is the same as the ID of the site in the log record $A_z^y$ (x=y) that means local access is made, then do nothing. Otherwise (x≠y) that means remote access is mode, then go to the following step.

Step 3: Calculate the total number of read/write accesses between the fragment $P_i$ and each accessing remote site $N_x$ respectively where x=1,2,3,..N.

Let $n_j^x$ be the total number of read/write accesses made to the fragment $P_i$ allocated at local site $N_x$ by different remote site $N_y$. If ($n_j^x$ <α ), then do nothing, otherwise to the following step.

Step 4: Calculate the total number of read accesses made to the fragment $P_i$ from all sites (including site $N_y$) within time interval β up to current access time t. Let $nr_j^x$ be the total number of read accesses made to the fragment $P_i$ allocated at site $N_x$ by site $N_y$ within time interval β up to current access time t.

Step 5: Calculate the average read volume of data transmitted (in bytes) between fragment $P_i$ and all the sites (including site $N_y$) from where accesses to the fragment $P_i$ are made within time interval β up to current access time t. Consider $A_z^y$ $Vr_i^x$ denotes the volume of read data transmitted (in bytes) between the fragment $P_i$ allocated at site $N_y$ and the site $N_x$ in the access $A_z^y$ log record. Furthermore let $Vr_i^x t$ denotes the average volume of read data transmitted between the fragment $P_i$ allocated at site $N_y$ and all the accessing site N occurred within time interval β up to current access time t, then:

$$Vr_i^x\ t = (\sum A_z^y\ Vr_i^x / \sum nr_i^{\ x}) \qquad (2)$$

Step 6: IF the average volume of read data transmitted of each accessing remote site ($Vr_i^x$) does not greater than the average volume of read data transmitted of all accessing site $Vr_i^x t$, then do nothing, otherwise go to the following step.

Step 7: If there is only one accessing remote site j qualify constraints stated in step 6. Calculate the volume of write data (in bytes) between the fragment $P_i$ and remote site $N_y$ and also calculate the volume of write data (in bytes) between the fragment $P_i$ and local site q $N_x$ within time interval β up to current access time t. The volume of write data can be calculated using equation – (3)

$$Vw_i^x t = (\sum A_z^y\ Vw_i^x) \qquad (3)$$

If $Vw_i^y t >$ $Vw_i^x t$ where x,y = 1,2,3, .... , N and $x \neq y$ , then the fragment $P_i$ is reallocated to site $N_x$ and removed from the current site $N_y$ and the fragment information matrix is updated accordingly.

Step 8: If more than one accessing remote sites qualify constraints stated in step 6. Calculate the volume of write data transmitted (in bytes) between the fragment $P_i$ and all qualify remote sites within time interval β up to current access time t and also calculate the volume of write data transmitted (in bytes) between the fragment $P_i$ and local site. Finally select the site which has the maximum update data volume than other sites and reallocate the fragment to that site. Then the fragment information matrix is updated accordingly. Otherwise, the update data volume of the two qualify sites is equal or a little different, and then go to the following step.

Step 9: Select the site which has the maximum distance from the current site and reallocate the fragment to that site and update the fragment allocation information.

## 4. Implementation Result

In this work, Product relation is used for our distributed database to test the proposed model. As horizontal fragmentation, Product relation is partitioned into 7 subsets of rows by use of the selection operator. Suppose that there are total seven fragments ($P_1$,$P_2$, ... ,$P_7$) of global relations in DDS. Let there are four sites in distributed database system placed at some distance respectively as following Table II.

The distance shown in Table II is in km. Then, sites $N_1$, $N_2$ are created into same group and respectively sites $N_3$, $N_4$ in group 2. All fragments of Product relation are initially allocated to the four sites in distributed database system like as Fig 2. 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. The entries of Log_Info table at site N1 at a particular point of time is shown in below Table IV..

Let Access threshold for fragment reallocation (α) is set to 2, and Time constraint for fragment reallocation (β) is set to 5 days. The total number of remote sites $N_3$ and $N_4$ access made to fragment $P_1$ within time constraint β up to current access time t (2018-04-26) in the Log_Info is 2. In the next day (2018-04-26), $n_{P1}^{N3}$ and $n_{P1}^{\ N4}$ is becoming 3 where $n_{P1}^{N3}$ and $n_{P1}^{N4} = 3 > \alpha$. Then, there are two sites qualify to be possibility migrate fragment $P_1$ and the transactions after 2018-04-26 are not taken.

**Table 2:** Site Distance Matrix

| Site | $N_1$ | $N_2$ | $N_3$ | $N_4$ |
|---|---|---|---|---|
| $N_1$ | 0 | 100 | 300 | 450 |
| $N_2$ | 100 | 0 | 200 | 350 |
| $N_3$ | 300 | 200 | 0 | 150 |
| $N_4$ | 450 | 350 | 150 | 0 |

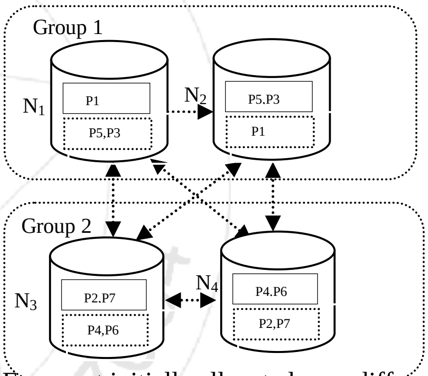


**Figure 2:** Fragment initially allocated over different sites

**Table 3:** Fragment Allocation Information

| Fragment | Primary Site | Copy Site |
|---|---|---|
| $P_1$ | $N_1$ | $N_2$ |
| $P_2$ | $N_3$ | $N_4$ |
| $P_3$ | $N_2$ | $N_1$ |
| $P_4$ | $N_4$ | $N_3$ |
| $P_5$ | $N_2$ | $N_1$ |
| $P_6$ | $N_4$ | $N_3$ |
| $P_7$ | $N_3$ | $N_4$ |

**Table 4:** LOG_INFO

| AFID | ASID | ADateTime | RorWAS | DataVol |
|---|---|---|---|---|
| $P_1$ | N1 | 2018-04-23 2:35:20 | r | 850 |
| $P_1$ | N3 | 2018-04-23 10:55:35 | r | 700 |
| $P_1$ | N4 | 2018-04-23 11:20:10 | r | 800 |
| $P_3$ | N1 | 2018-04-24 18:11:21 | r | 650 |
| $P_1$ | N4 | 2018-04-24 20:15:35 | w | 310 |
| $P_1$ | N2 | 2018-04-25 10:12:34 | r | 125 |
| $P_1$ | N3 | 2018-04-26 19:00:39 | w | 300 |
| $P_1$ | N4 | 2018-04-26 20:11:14 | r | 750 |
| $P_1$ | N3 | 2018-04-26 15:30:43 | r | 800 |
| $P_3$ | N1 | 2018-04-27 20:12:24 | w | 550 |
| $P_1$ | N3 | 2018-04-27 13:24:43 | w | 700 |

The next step, Using equation (2), the average volume of read data transmitted between fragment $P_1$ and all sites which accessed fragment $P_1$ , $Vr_{P_1}^{Nx}t$ is (850+700+800+125+750+800) bytes / 6 = 680.8 bytes where x=1,2,3,4. Next, calculate the average volume of read data transmitted between fragment $P_1$ and two sites which accessed fragment $P_1$ , $Vr_{P_1}^{N3}$ by using equation (l) as: = (750 + 800) bytes / 2 = 750 bytes, $Vr_{P_1}^{N4}$=(800+ 750) bytes / 2 = 775 bytes. Since 750 bytes, 775 bytes > 680.8 bytes, $n_{P_1}^{N3}$ and $n_{P_1}^{N4}$= 3> α respectively, therefore fragment $P_1$ need to be reallocated to sites $N_3$ and $N_4$ which are qualified.

Now calculate the volume of write data of sites $N_3$ and $N_4$ accessed made to fragment $P_1$ by using equation(3) as: 300 bytes and 310 bytes where the volume of write data (310 bytes )of site $N_4$ is a little greater than the volume of write data (300 bytes) of site $N_3$. Therefore it is a little different and the distance of the two sites need to be considered. Finally, select the site $N_4$ which has the maximum distance from the current site and the site $N_4$ which has the maximum volume of write data accessing the fragment $P_1$ allocated at site $N_1$ rather than the site $N_3$.Next, fragment $P_1$ is migrated to the site $N_4$ and then the copy of $P_1$ is also migrated to the site $N_3$ that is the same group $N_4$ and updates the fragment allocation information. After migration process from every four sites is finished, the fragment allocation information is update and the fragment is reallocated in Figure 3.

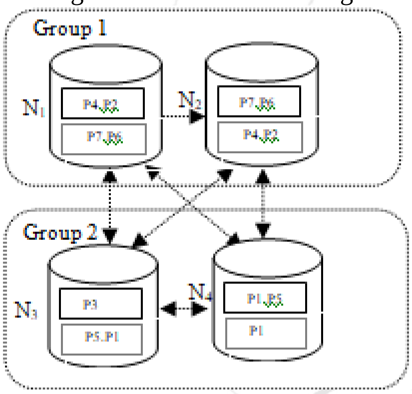


**Figure 3:** Fragment reallocated over different sites

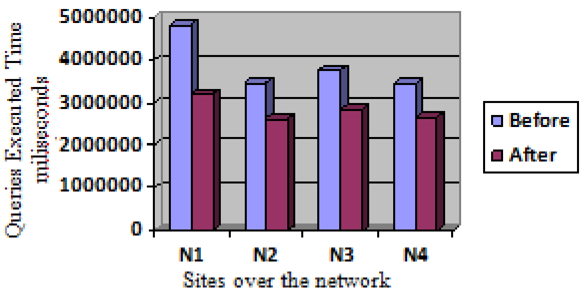


**Figure 4:** Executed Time of queries for four sites after and before applied proposed algorithm

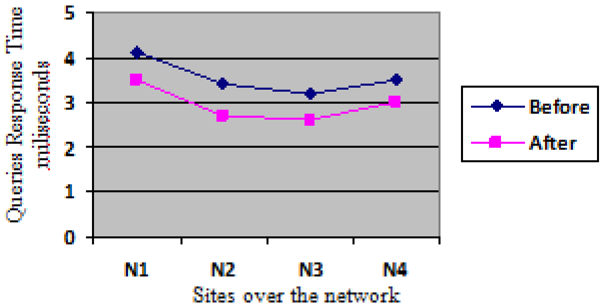


**Figure 5:** Response Time of queries for four sites after and before applied proposed algorithm.

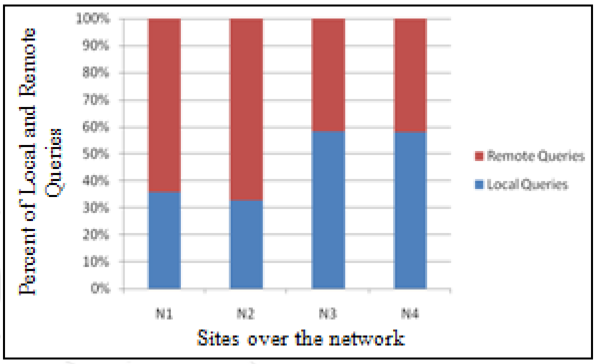


**Figure 6:** Percent of Local and Remote Queries for Four Sites before Applied Proposed Algorithm

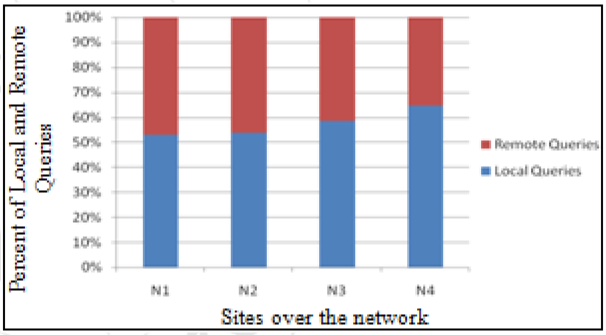


**Figure 7:** Percent of Local and Remote Queries for Four Sites after Applied Proposed Algorithm

In the experimental result, Dell PC of core i5 processor with 4 GB RAM and 1 GB graphic card is used. Each site generate a set of queries which has the number of queries is 10000 and 80 percent of retrieval queries and 20 percent of update queries. Access threshold value for fragment reallocation is set to 15 percent of the number of queries and 1 week for Time constraint for fragment reallocation. The result shows in Figure 4 that the executed time of queries is significantly difference in site 1 or 3 after migration. In figure 5, the response time of queries for four sites after finishing migration process is slightly better than response time of queries before finishing migration. After applied the proposed algorithm, the percent of remote queries for each site decrease and the percent of local queries increase for each site respectively in figure 7.

## 5. 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 algorithm for partial-replicated dynamic fragments allocation in distributed database system is presented. In fragmentation and replication phase, horizontal fragmentation algorithm can reduce the irrelevant data access by the other remote sites and increase the local control access. In migration phase, when there are two site qualified for fragment reallocation , the proposed algorithm selects the site which has the maximum volume of update data accessing the fragment and respectively the site contains maximum distance than another one site. Therefore, the proposed algorithm will reduce update cost. But this algorithm will use more space than TTVDCA because it store copy of some fragments in each site. 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.

## 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] 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.
[3] Arjan Singh and K. S. Kahlon, "Non-replicated Dynamic Data Allocation in Distributed Database Systems", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.9, September 2009.
[4] 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.
[5] M. Oasu and P. Valduriea, "Principles of Distributed Database Systems". Prentice Hall. Second ed. 1999.
[6] Rababaah, H. Spring 2005. Distributed Databases Fundamentals And Research. Advanced Database – B561 Dr.H. Hakimzadeh ,Department of Computer and Information Sciences Indiana University South Bend.
[7] Shahin Kamali, Pedram Ghodsia, Khuzaima Daudjee, David R. Cheriton School of Computer Science. "Dynamic Data Allocation with Replication I Distriubted Systems", 2100 IEEE
[8] S. March and S. Rho, "Allocating Data and Operations to Nodes in Distributed Database Design," IEEE Trans. Knowledge and Data Eng., vol. 7, no. 2, pp. 305-317, 1995.
[9] I.O. Hababeh, M. Ramachandran and N. Bowring, "A high-performance computing method for data allocation in distributed database systems", Springer, J Supercomput (2007) 39:3-18.
[10] 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.
[11] A. Mei, L. Mancini, and S. Jajoda, "Secure Dynamic Fragment and Replica Allocation in large-Scale Distributed File Systems", IEEE Trans. Parallel and Distributed Systems, vol.14, no.9, pp. 885-896, Sept.2003.
[12] Raju Kumar, Neena Gupta, "Non-Redundant Dynamic Data Allocation in Distributed Database Systems", Special Issue of International Journal of Computer Applications, on Issues and Challenges in Networking,Intelligence and Computing Technologies - ICNICT, pp. 6-10, November 2012.
[13] Y. Huang and J. Chen, "Fragment Allocation in Distributed Database Design," J. Information Science and Eng., vol. 17, pp. 491- 506, 2001.
[14] A. Tamhankar and S. Ram, "Database Fragmentation and Allocation: An Integrated Methodology and Case Study," *IEEE Trans. Systems, Man and Cybernetics*—Part A, vol. 28, no. 3, May 1998.
[15] K. Karlaplem and N. Pun, "Query-Driven Data Allocation Algorithms for Distributed Database Systems," Proc. Eighth International Conf. Database and Expert Systems Applications (DEXA '97), pp. 347-356, Sept. 1997.
[16] O. Wolfson, S. Jajodia, and Y. Huang, "An Adaptive Data Replication Algorithm," ACM Trans. Database Systems, vol. 22, no. 2, pp. 255-314, 1997.
[17] G. Chiu and C. Raghavendra, "A Model for Optimal Database Allocation in Distributed Computing Systems," Proc. IEEE INFOCOM 1990, vol. 3, pp. 827-833, June 1990.
[18] 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
[19] 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.

## Author Profile

**Nang Khine Zar Lwin** received the Bachelor and Master degrees in Computer Sciences from University of Computer Studies (Kyaing Tong) in 2004 and 2010, respectively. She is an assistant lecturer from University of Computer (Kyaing Tong), Myanmar. Now, She is also a Ph.d candidate from University of Technology (Yadanarpone Cyber City), Pyin Oo Lwin, Myaanmar.