Non-Redundant Dynamic Fragment Allocation with Horizontal Partition in Distributed Database System

Nang Khine Zar Lwin
Faculty of Information & Communication Technology
University of Technology (Yadanarpon Cyber City)
Pyin Oo Lwin, Myanmar
nangkhinezar@gmail.com

Dr.Tin Myint Naing
Faculty of Information & Communication Technology
University of Technology (Yadanarpon Cyber City)
Pyin Oo Lwin, Myanmar
utinmyintnaing08@gmail

***Abstract*— The main task of distributed database is how to fragment the global database into small fragments, how to allocate and replicate the fragments among different sites over the network. The performance of the distributed database system can be increased according to the best way of fragmentation, allocation and replication. Dynamic fragment allocation technique provides many environments where access patterns of different sites from multiple locations made to fragment change over time. This paper proposes an approach for non-redundant dynamic fragment allocation in distributed database system which additionally modified read and write data volume factor to Threshold Time Volume and Distance Constraints Algorithm. The proposed approach reallocates fragments with respect to the access patterns made to each fragments with amount of data volume up to time constraint and threshold value. The write data volume has to be considered for relocation process when more than one site simultaneously qualifies for the fragment. This algorithm will improve the overall of distributed database system performance.**

Keywords— Distributed Database System; Fragmentation; Dynamic Fragment Allocation;

I. Introduction

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. The different geographic of a computer network can communicate through a network and it is managed by a distributed database management system [3]. 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. In distributed database system, the task of allocation fragment plays a prominent role in all aspects of the system. The aim of a fragment allocation algorithm is to determine which fragment is properly located to which sites so as to minimize the total data transfer cost and reduce the executed time of queries.

The most studies of data allocation focus on which fragment to locate which site. To achieve the best performance of the system, fragments should be stored near site where they are most frequently access. The performance and data availability of system depend on data allocation strategies [4]. The fragment allocation algorithm can be solved the problem on static access pattern and dynamic access pattern from many different sites. In distributed database, there are three types of fragmentation strategies: horizontal, vertical and mixed. Sub-divides attributes of the relations into fragments called vertical fragments. Sub-divides types into fragments, all having the same attributes of the original relation called horizontal fragmentation. Mixed fragmentation refers to a combination of horizontal and vertical strategies.

In this paper, it focuses on approach of non-replicated dynamic fragment allocation algorithm in distributed database system. The proposed algorithm is based on TTCA algorithm by A. Singh, and K.S.Kahlon [5]. This approach already includes access threshold, access time constraints. In this algorithm, if the access time of one site to a fragment exceeds the threshold value during time constraints, fragment will reallocate to the site which frequently access to fragment. The volume of data transmission is additionally added by N.Mukharjee [2]. Raju Kumar, Neena Gupta [1] proposed an extended approach to non-replicated dynamic fragment allocation in distributed database system. The proposed algorithm is additions of Threshold Time Volume and Distance Constraints Algorithm [1] by analyzing data read and write operation instead of data volume only and reallocate the related fragments dynamically to closer site. This algorithm solved the problem when there are more than one site qualifies for the fragment by choosing the site which has the maximum volume of write data. The algorithm is proposed for non-redundant dynamic fragment allocation in distributed database system.

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 non-redundant dynamic fragment allocation in distributed database system. Section IV explains the result of implementation of the system model. Section V describes the comparison of the proposed algorithm and other non-redundant algorithms. Section VI summarizes the contribution of the study.

II. Related Work

Many researchers have been done on the problem of fragment allocation to different sites. Several data allocation approaches based on static data allocation strategy has been

978-1-5386-7516-8/18/$31.00 ©2018 IEEE

published over past few years. 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 [20]. A machine learning approach had adapted the approach for allocating fragments was provided by [15]. An approach based on lagrangian relaxation was provided in [21] and heuristic approaches have been explained by [16]. Moreover, a mathematical modeling approach and a genetic algorithm base approach to allocate operation to nodes have been presented by [17]. A high-performance computing method for data allocation in distributed database system is proposed by [19]. Several data allocation approaches based on static data allocation strategy has been provide in recent years.

Most of recent researches have been published dynamic data allocation algorithm in database system over past few years. Dynamic data allocation algorithm is proposed by [23] which reallocate data with respect to changing data access pattern. An optimal algorithm for non- replicated database system is proposed by [6]. 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. In [15] threshold algorithm for non- replicated distributed database was proposed. This algorithm reallocated the fragments respect with the changing data access patterns. In [2] 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 [22] Extended Threshold Algorithm was developed which solved the scaling problem and decreased the space requirement as time constraint is not stored. In [2] Threshold Time and Volume 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 [2,5,12,18] that it is more than one site quality for the fragment relocation by selecting the site which occupy the fragment.

Extended dynamic fragment allocation algorithm is proposed by [1] which is the extension of [2] by adding distance factor besides existing factors to solve the problem in [2,5,12,18]. Threshold Time Volume and Distance Constraints Algorithm use extra storage cost according to storing corresponding row of site distance matrix at each site. If there is more than one site quality for the fragment relocation, it selects the site which is at maximum distance from current site. It only considers for the distance factor, it needs to be considered other factors rather than this distance factor.

In this paper, an approach to non-redundant dynamic fragment allocation with horizontal partition algorithm is proposed. Fragments is migrated to the site which has maximum read and write data volume with the access count made to the fragment exceeds threshold value during time constraints. To solve the problem in [2,5,12,18] when more than one site qualified to the fragment, reallocation processes migrate the fragment to the site which has maximum write data volume made access to fragment among other different sites. This algorithm will improve the overall performance of the system and reduce update cost of sites.

III. The Proposed Non-Redundant Dynamic Fragment Allocation with Horizontal Partition Algorithm

The proposed algorithm can be considered into two phases: preprocessing phase and action phase. In the processing phase, data collection and preprocessing, fragmentation and allocation processes are performed before accessing the fragment. In the action phase, the processes of this phase are performed every time after each access to the fragment. The overall procedure of the proposed model is shown in Fig.1.

A. Data Collection and Preprocessing

In this section, sample database is collected as the first step. In preprocessing process, assumed that there is a fully connected network consisting of many different sites S= $\{s_1, s_2, s_3, ..., s_n\}$. Access threshold value and time constraint have to be specified in this step. Each site stores log information table (identifier of the fragment accessed, name 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 includes some information for each fragment access to that site assigned in each site. The used notations in this paper are described in Table I.

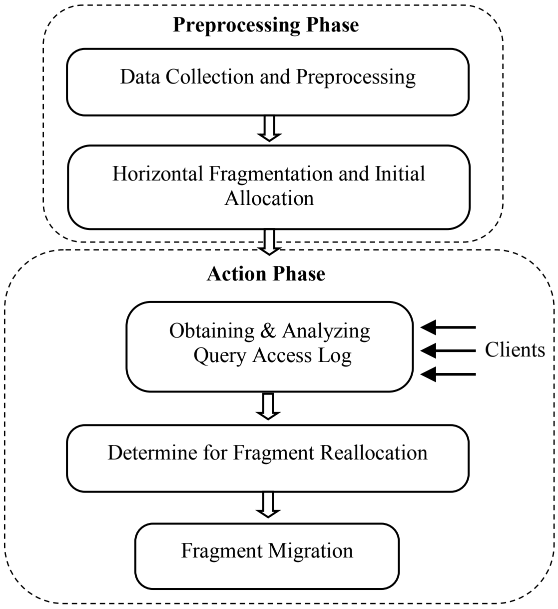

Fig. 1. The overall procedure of proposed algorithm

TABLE I. Algorithm Notations

Notation	Meaning
R	Global relations in distributed database system

F	Number of fragments of global relations in distributed database system
S	Number of sites in distributed database system
f_i	The i^{th} fragment
s_n	The n^{th} site
α	Access threshold value
β	Time constraint
$A_z{}^y$	Log Information record z^{th} access at site y
N_i^x	Total number of read and write accesses from site s_x to the fragment f_i within time interval β up to current access time t
V_i^x	Volume of read and write data transmitted between fragment f_i and site s_x within time interval β up to current access time t
Vw_i^x	Volume of write data between fragment f_i and site s_x within time interval β up to current access time t

B. *Horizontal Fragmentation and Initial Allocation*

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

$$f_j = {}_{\sigma Qj}(R), \text{ where } 1 \le j \le i \quad (1)$$

Where Q_j is the selection conditions as a simple predicate and i is the maximum number of fragments. In horizontal fragmentation, the base table can be reconstructed because each fragment contains a subset of the rows in the table. Every horizontal fragment must have all columns of the original base table. Suppose that global relation is partitioned into small fragments or partitions using equation (1) such that R = {f_1, f_2,..., f_i}. 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 allocation refers to the allocation of data at multiple sites served by a computer network. Each Fragment must be allocated to location in the distributed environment such that the system functions effectively and efficiently. Fragment allocation can further be divided into two different methods: redundant and non-redundant. In non-redundant allocation exactly one copy of each fragment will exist across all the sites, while under a redundant allocation, fragments are replicated over multiple sites.

Some requirement and initial allocation process are performed in this phase. Suppose that there are F fragments of global relations and S sites in the distributed database system. All the small fragments of global relation are initially allocated over different sites using any static allocation method in non-redundant manner. Each site contains at least one fragment. Each site contains Log_Info table and pre-specified access threshold value and pre-specified time constraint. The structure of log information 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 Log_Info 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 s_y, where z=1,2,3, ..., ∞ and y = 1,2,3,...,S)

C. *Obtaining and Analyzing Queary Access Log*

In this process, Log_Info 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 f_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). Query traces that are invoked from many clients via some applications are simplified to obtain read or write operation. The local agent performs the following steps 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: If the ID of local site s_y 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 made, and then go to the following next process.

D. *Determine Fragment Reallocation and Fragment Migration*

In reallocation process, the proposed model selects the site which has maximum average data volume access to fragment within time interval β up to current access time t and also selects the site which has maximum write data volume when more than one qualified sites are not significantly different. In this process, the following steps must be done within time interval β up to current access time t. The volume of data is considered in bytes.

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

Step 2: Calculate the average volume of read and write data transmitted between fragment f_i and all the sites (including local site s_y) that made access to the fragment f_i. Consider $A_z{}^y$ V_i^x denotes the volume of read and write data transmitted between the fragment f_i allocated at site s_y and the site s_x in the access $A_z{}^y$ log record. Furthermore let $V_i^x t$ denotes the average volume of read and write data transmitted between the fragment f_i allocated at site s_y and s_x made access to the fragment f_i occurred within time interval β up to current access time t then:

$$V_i^x t = (\sum A_z^y V_i^x / \sum N_i{}^x) \quad (2)$$

Step 3: If the average volume of read and write data transmitted of each accessing remote site does not greater than the average volume of read and write data transmitted of all accessing site, then do nothing, otherwise go to the following step.

Step 4: If there is only one accessing remote site s_x qualify constraints stated in step 3, then the fragment f_i is reallocated to site s_x and removed from the current site s_y and the fragment allocation information matrix is updated accordingly.

Step 5: If more than one accessing remote sites qualify constraints stated in step 3. Calculate the volume of write data transmitted between the fragment f_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 – (3)

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

Finally, select the site which has the maximum write data volume than other sites and reallocate the fragment to that site. Then the fragment allocation information matrix is updated accordingly. In practical result, the difference between read and write query executed time is not equal and is shown in fig.2 and fig.3.

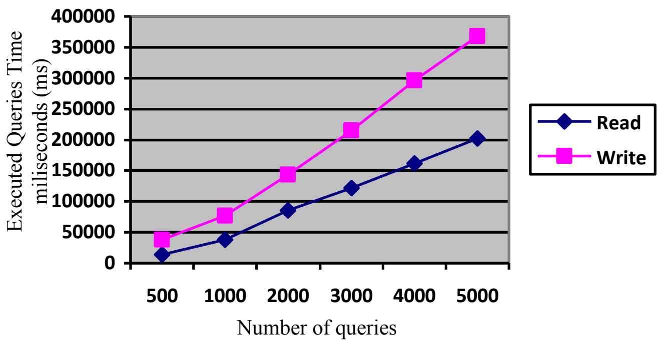

Fig. 2. Difference between read and write executing query time in local node

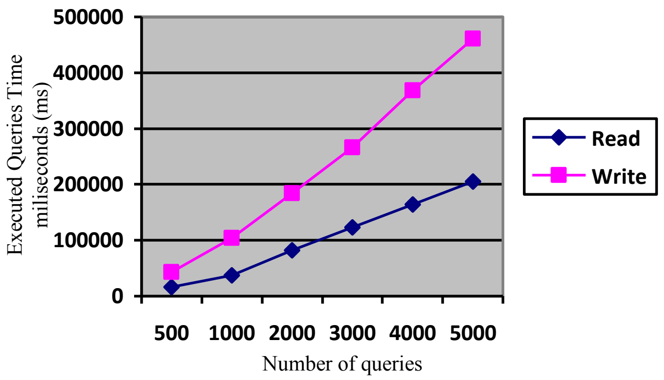

Fig. 3. Difference between read and write executing query time in remote node

Therefore, the new factor of read and write query is necessary to consider for fragment reallocation. In our new algorithm, the new factor is need to consider when more than one site qualify for fragment reallocation.

IV. Implementation Result

In this study, there are four sites in fully connected distributed database system that is S= $\{s_1,s_2,s_3,s_4\}$. In preprocessing phase, one of sample global relation is used for our distributed database to test the proposed algorithm. Let access threshold for fragment reallocation (α) is set to 3, and time constraint for fragment reallocation (β) is set to 5 days. Each site stores access log information. The global relation(R) is partitioned into eight fragments using horizontal fragmentation (equation 1) such that R= $\{f_1,f_2,f_3,\ldots,f_8\}$. All fragments of global relation are initially allocated to the different four sites in distributed database system as shown in Fig 4. These fragments are allocated in non-redundant manner over different four sites as shown in Table II. 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.

In action phase, queries invoked from many clients are obtained and stored into Log_Info table at site s_1 at a particular point of time are shown in below Table III. The total number of remote sites s_2 and s_3 access made to fragment f_1 within time constraint β up to current access time t (2018-07-27) in the Log_Info is 4 > α. Then, there are two sites qualify to be possibility migrate fragment f_1.

TABLE II. FRAGMENT ALLOCATION INFORMATION

Fragment	Site
f_1	s_1
f_2	s_4
f_3	s_3
f_4	s_2
f_5	s_1
f_6	s_4
f_7	s_2
f_8	s_3

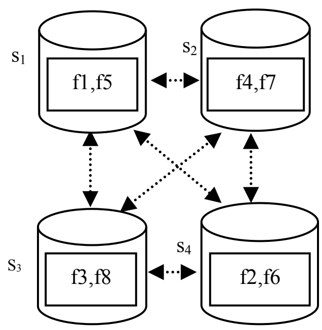

Fig. 4. Fragment initially allocated over different sites

TABLE III. LOG_INFO

AFID	ASID	ADateTime	RorWAS	DataVol
f_1	s_1	2018-07-23 2:35:20	r	350
f_1	s_2	2018-07-23 10:55:35	r	600
f_1	s_3	2018-07-23 11:20:10	r	700
f_1	s_2	2018-07-24 18:11:21	r	500
f_1	s_2	2018-07-24 20:15:35	w	250
f_1	s_3	2018-07-25 10:12:34	r	300
f_1	s_2	2018-07-26 19:00:39	w	400
f_1	s_3	2018-07-26 20:11:14	w	320
f_5	s_4	2018-07-26 15:30:43	r	200
f_1	s_3	2018-07-27 20:12:24	r	450
f_5	s_3	2018-07-27 13:24:43	r	300

In determine fragment reallocation process, using equation (2), the average volume of read and write data transmitted between fragment f_1 and all sites which accessed fragment f_1 , $V_{f1}{}^{sx}t$ is (350+600+700+500+250+300+400+320+450) bytes / 9 = 430 bytes where x=1,2,3,4. Next, calculate the average volume of read and write data transmitted between fragment f_1 and two sites which accessed fragment f_1 , $V_{f1}{}^{s2}$ by using equation (2) as: = (600+500+250+400) bytes / 4 = 438 bytes, $Vf_1{}^{s3}$ = (700+300+320+450) bytes / 4 = 443 bytes. Since 438 bytes and 443 bytes > 430 bytes, $N_{f1}{}^{s3}$ and $N_{f1}{}^{s4}$ = 4> α respectively, therefore fragment f_1 need to be reallocated to sites s_2 and s_3 which are qualified. Now calculate the volume of write data of sites s_2 and s_3 accessed made to fragment f_1 by using equation(3) as: 250bytes and 320 bytes where the volume of write data (320 bytes)of site s_3 is greater than the volume of write data (250 bytes) of site s_2.

Since Fragment f_1 is migrated to the site s_3 which has the maximum volume of write data access to the fragment f_1 allocated at site s_1 rather than other sites. Transferring fragment f_1 to site s_3 which has the maximum average volume of write data of the accessed fragment causes f_1 that results faster access and reduces update cost. After migration process from every four sites is finished, the fragment allocation information is updated and fragments are reallocated as shown in Fig. 5.

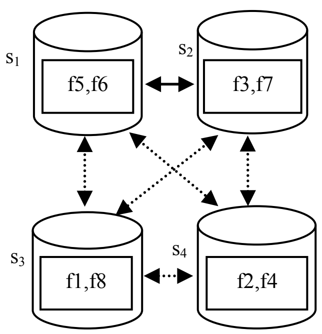

Fig. 5. Fragment reallocated over different sites

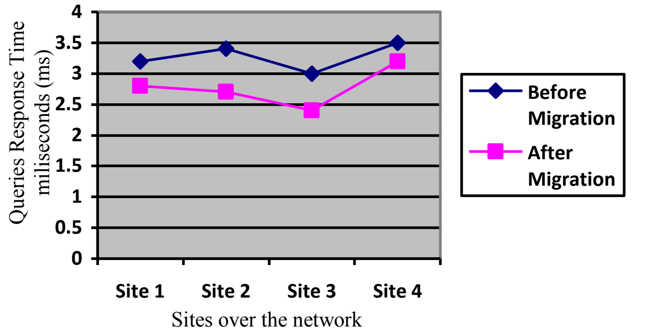

Fig. 6 Response Time of queries for four sites after and before applied proposed algorithm

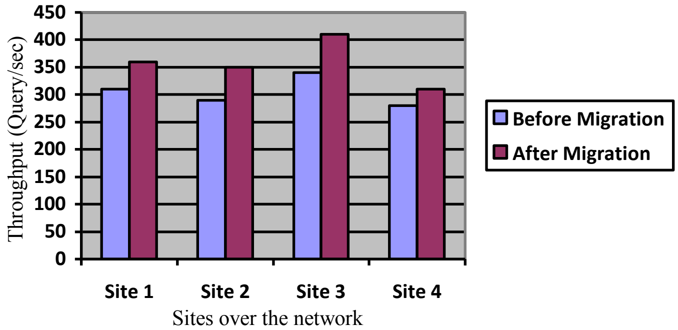

Fig. 7 Throughput comparisons of after and before applied proposed algorithm

In the above result, there are made in Dell PC of core i5 processor with 4 GB RAM and 1 GB graphic card. We create a set of queries which characterize the fragments' access from each different four sites. For each sites generate a set of queries which has the number of queries is 10000 and 75 percent of retrieval queries and 25 percent of update queries. Before reallocation process is performed, we run the set of queries from different sites as many clients use the application. We set Access threshold value for fragment reallocation is 25 percent of the number of queries and 1 week for time constraint for fragment reallocation. And after the reallocation process has finished, we run the set of queries again. The result shows in Fig 6 that the response time is not so difference in site 4, but it is significantly difference in more sites as a result of after migration. The throughput comparison of four sites before migration process and after finishing migration process in Fig. 7. The result shows that the throughput of four sites after migration is slightly better than the throughput of four sites before migration.

V. COM PARISON

The comparison of the proposed algorithm in theoretically with four parameters: update cost, storage cost, handling of multiple sites simultaneously qualified for fragment migration and response time. The cost in executing queries depends on the data transfer cost from different sites. Moreover, the cost of updating fragment could be higher than the cost of reading fragment and response time may be slower where update queries are executed on remote site.

The proposed algorithm selects the site which has the maximum volume of write data accessing the fragment when more than one site simultaneously qualified for fragment migration. Therefore, the proposed algorithm reduces update cost and also reduces storage cost than TTVDCA because the proposed algorithm does not need to store corresponding row of Site Distance Matrix at each site. When there are two sites qualified for fragment migration, TTVDCA reallocate the fragment which is at maximum distance from the current site where fragment is allocated. This algorithm selects further site according to improve response time for the site has to travel than other site. This proposed algorithm can also solve the problem that is occurred when more than one site qualified for fragment reallocation in threshold, optimal, TTCA and ETA algorithm. In summary, the proposed algorithm is better than other non-replicated algorithms.

VI. CONCLUSION

The allocation of fragment over different sites is important on performance of distributed database system. In this study, we present the algorithm for non-redundant dynamic fragment allocation in distributed databases system. In fragmentation process, horizontal technique is use to fragment global relation into small fragments to reduces the amount of irrelevant data accessed by the transactions of the database, thus reducing the number of disk accesses.

The proposed algorithm calculates the volume of read and write data amount to be considered for fragment reallocation process and reallocate fragment to the site that has the frequently write access intend to reduce total transfer cost from different sites and total access time. In future work, other factors need to be considered besides already exiting factors and the algorithm can be provided in redundant algorithm.

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] 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

[3] M. Oasu and P. Valduriea, "Principles of Distributed Database Systems". Prentice Hall. Second ed. 1999.

[4] 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.

[5] 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.

[6] W.W.Chu "Opatimal File Allocation in Multiple Computer Systems" IEEE Transaction on Computer, Vol.C-18,No.10,1969.

[7] 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.

[8] Shahin Kamali, Pedram Ghodsia, Khuzaima Daudjee, David R. Cheriton School of Computer Science. "Dynamic Data Allocation with Replication I Distriubted Systems", 2010 IEEE

[9] 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.

[10] P. Apers. "Data Allocation in Distributed Database", ACM Trans. Database System, vol. 13, no.3, pp. 263-304, Sept 1988.

[11] Fardin Esmaili Sangari, Seyed Mostafa Mansourfar "A New Dynamic Data Allocation Algorithm for Distributed Database". International Journal of Computer Science and Information Security Vol. 9, No. 5, May 2011, pp. 138-141.

[12] 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.

[13] 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.

[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] 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.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] Raju Kumar, Neena Gupta, "Non-Redundant Dynamic Data Allocation in Distriubted Database Systems", Special Issue of International Jouranl of Computer Applications, on Issues and Challenges in Networking,Intelligence and Computing Technology, ICNICT, pp.6-10, November, 2012.

[23] 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.