An Extended Technique for Data Partitioning and Distribution in Distributed Database Systems (DDBSs)

Ali A. Amer[1], Adel A. Sewisy[2]
Computer Science Department, Science College, Taiz University, Yemen[1]
Computer Science Department, Computer Science College, Assuit University, Egypt[2]
aliaaa2004@yahoo.com[1]; adel.mohamed@fci.au.edu.eg[2]

Abstract— **In this work, an extended heuristic technique for horizontal data partitioning and distribution is intelligently evolved. As a matter of fact, introducing a practical solution for DDBS rendering enhancement is the major focus of this work. Meanwhile, such an intended enhancement is bound to be achieved through presenting clever data partitioning method, drawing outstandingly-designed site clustering algorithm, and developing mathematically-calculated data allocation cost model. Nevertheless, as partitioning technique has already been developed, this work comes to further extend this technique by having it skillfully incorporated with newly-proposed site clustering algorithm and mathematical model for data allocation, including data replication, for the sake of precisely producing super effective comprehensive technique. Consequently, this proposed technique is certainly set to be promising and capable of tremendously reducing the overall costs of data transmission (TC). In the sense that it is believed that such significant extension is to overwhelmingly be a potential progress of profoundly beneficial effects on overall DDBS performance.**

Keywords— Horizontal; Partitioning; Allocation; Replication; Site Clustering; DDBS.

I. Introduction

It is undeniable that Distributed Database systems (DDBSs) become an increasing demand for most aspects of our technology-based life. Subsequently, the need to greatly-appreciated design, on the long term, for DDBS still bubbles to surface as it has a leading impact on DDBS productivity. Off the most important challenges still need to be carefully tackled is Transmission Costs (TC). In DDBS, on the other hand, there are several methods by which TC could be tremendously mitigated. Among these methods are: data clustering algorithm (partitioning), data placements strategies, and network site clustering techniques. Therefore, this work comes to integrate some of these methods/techniques into a single efficient work in the purpose of optimizing work proposed in [1]. For site clustering, a hierarchical-inspired clustering algorithm is presented. It is worth indicating that integrating site clustering would definitely come with remarkable benefits in terms of TC reduction as shown in [2]. Moreover, mathematical cost model is set to be given in the sake of paving the way to find much more efficacious data allocation (and replication) model. . In short, the contribution of this work are clearly listed as follows;

1. Attentively amending the objective function drawn in [1] that transmission cost (including query costs) is significantly reflected [2].
2. Drawing hierarchical clustering algorithm for network sites.
3. Mathematically formulating data allocation and replication model as it had not been given in [1]. It is worth indicating that the proposed data allocation model is meant to be applicable in both works as it is being done to be completely complied with proposed work's modifications including sites grouping.
4. Presenting an illustrative step-by-step explanation along with brief experimental results for both works of [1] and present work of tis paper, in a clear way in purpose of obviously showing behaviors of both works.

The rest of this paper is structured as follows; section (II) explores earlier studies which are closely relevant to this work. In section III, technique's methodology including architecture and objective function is briefly addressed. Algorithm of site clustering is stated in section IV. In section V, the proposed data allocation and replication models are clearly presented. In section VII, to proof the concept, a simple briefing for experimental results for one single experiment are vividly delineated. Lastly, conclusions and future work directions are drawn in section VIII.

II. Literature Survey Review

Given the importance of DDBS, a considerable number of Horizontal Partitioning (HP) methods have been proposed in literature in consecutive steps just to improve DDBS performance. For instance, in [3, 4], a min-term predicate was used as metric to divide relations so that primary HP was produced providing that previously-determined predicates set supposedly met the disjoint-ness and completeness properties. While [5] presented two-phase horizontal partitioning. Relations were first partitioned by primary horizontal partitioning using predicate affinity the bond energy algorithm; followed by further partitioning using derived horizontal partitioning. In [6], Create, Read, Update and Delete Matrix (CRUD) was proposed to design DDBS at the initial stage. Relation attributes used as rows of CRUD and applications locations used as columns. Additionally, data allocation was considered as well.

To find an optimal horizontal partitioning, [7] proposed cost model so that two scenarios for data allocation were addressed that no supplemental complexity was added to data allocation. This model was professionally extended and mathematically shown to be an effective at reducing communication costs [1]. For reducing database access time, a hybridized partitioning is proposed in [8] based on subspace clustering algorithm to generate data partitions with respect to tuple and attribute patterns so that the closely correlated data were grouped together. Experimental results demonstrated that this clustering-based method were better in diminishing access time. Meanwhile, to increase data locality, [9] proposed a decentralized approach for dynamic table partitioning and allocation in DDBS (DYFRAM) based on the history of recorded access. Approach feasibility was hypothetically and experimentally demonstrated. For the same goal, to improve DDBS performance through increasing local accesses at run time over cloud environment; [10] presented an improved system to perform initial-stage partitioning and data allocation along with replication. Site clustering technique was addressed as well.

On the other hand, data allocation problem in DDBS was investigated in [11]. Two algorithms were developed with the aim of lessening the overall communication costs. By the same token, to draw queries behavior in DDBS; [12] presented two heuristic algorithms to find an optimal or near-optimal allocation scenario in terms of communication costs reduction. Compared to [11], this algorithms was successfully shown to be close enough from being an optimal. Meanwhile, [13] presented dynamic data allocation method to mainly decrease transmission costs, considering database catalog as the only storing place for required data as method was implemented. As a new of its kind, [14] sought to give partial data reallocation and full reallocation heuristics to minimize costs and maintain complexity minimized. Furthermore, in purpose of finding an optimal data allocation technique; in [15], a non-replicated dynamic data allocation approach was carefully developed. This algorithm (called, POEA) was originally aimed at integrating some previously-proposed concepts used in its earlier counterparts including [16]. [17], on the other hand, proposed a dynamic non-replicated data allocation algorithm (named, NNA), with respect to the changing pattern of data access along with time constraints, data reallocation was done.

In the meantime, [18] demonstrated data allocation framework for non-replicated dynamic DDBS using threshold [19], and time constraint [20] algorithms. This work was illustrated to be more effective in terms of long-term higher performance than threshold algorithms chiefly as access frequency pattern changes rapidly. However, [21] gave an extended data allocation approach which was capable of placing partitions dynamically in redundant/non-redundant DDBS. Moreover, problem of having more than one deserve-to-receive-data site was addressed. Finally, in [22], Data Replication Problem (DRP) was formulated to perform an accurate horizontal partitioning of overlapping partitions. This work sought to place N-copy replication scheme of partitions into M distinct sites with ensuring of removing data overlapping. To achieve such goal, replication problem was treated as an optimization problem so that partitions' copies and sites kept at minimum. This work however has further been extended in [23]. While a novel soft data locality constraints based on partitions' affinity was developed, DRP problem was re-formalized as an integer linear program. Data insertion and deletion were considered and runtime performance was analyzed as well.

III. Proposed Technique

In this technique, for partitioning phase, all requirements, heuristics, definitions, notations, and formulas drawn in [1] are all strictly used. The extension part however is set to be made through amending objective function as drawn in section (A) below, as well as incorporating both newly-proposed site clustering algorithm and mathematically-designed data allocation and replication models (Figure 1).

A. Objective Functions

$$TC_R = \sum_{j=1}^{m}\sum_{i=1}^{m}\sum_{k=1}^{q}(1 - X_{kj}) * (RF_{kj}) * P_{size} * CMS_{ij} \qquad 1$$

$$TC_U = \sum_{j=1}^{m}\sum_{i=1}^{m}\sum_{k=1}^{q}(1 - X_{kj}) * (UF_{kj}) * P_{size} * CMS_{ij} \qquad 2$$

$$TC_{total} = TC_R + TC_U \qquad 3$$

Where CMS_{ij} is expressed as follows;

$$CMS_{ij} = \begin{cases} CCM & \text{whenever costs considered between clusters of sites} \\ CSM & \text{whenever costs considered between sites} \end{cases}$$

And P_{size} is the size of partition under consideration.

For step eight of heuristics of [1], as each partition is set to be placed into clusters of sites, two scenarios would be considered. While in first scenario, the triggered partition would be replicated over all clusters. In second scenario, partition is to be assigned to clusters of partition's maximum costs. As to sites, on the other hand, whenever Average of Update Cost (AUC) is greater than Average of Retrieval Cost (ARC), the triggered partition is to be assigned to the site of maximum update cost inside its relative cluster providing that cluster's/site's constraints have not been violated. If constraints violation happens to be recorded, then partition would be assign to site of the next highest AUC inside the same cluster. On the contrary, for each partition, whenever ARC is greater

ISSN 2457-905X

than AUC, that partition is to be allocated to all sites requesting it as it is being done in [1].

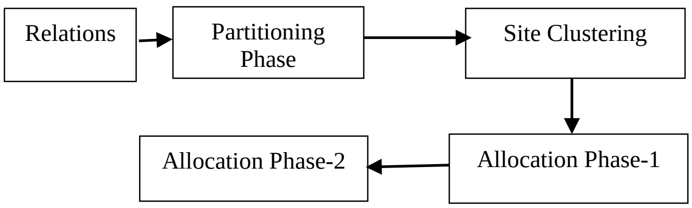

Figure 1: The Proposed System Phases

IV. Site Clustering

The presented algorithm of site clustering has been made based on concept of hierarchical clustering, especially as initial clusters are built. Then, clustering is to be entirely kept proceeding up based on the least average of communication costs between sites to decide site's belonging as site being taken to be grouped. It goes without saying that as network sites being clustered, the communication costs within and between clusters are of key importance to be taken for data allocation phase particularly in the non-replication scenario [2]. In the meantime, the symmetry average of communication costs would be used as it has been proved to be rapid, reliable and an efficient method [3; 24]. In the sense that the cost matrix is assumed to be a symmetric between sites/clusters and cost between the same sites/clusters is considered to be a zero or.

V. The Proposed Allocation Model

A. Problem Description

In DDBS, it has long been proved that the optimal solution to promote DDBSs performance is set to be done by properly partitioning data, and carefully allocating partitions into cluster/site in where it is mostly accessed [25]. This problem, on the other hand, counts deeply on the complexity embedded in choosing cluster/site for targeted data. In fact, one solution is believed to highly contribute in achieving intended performance; so that the number of update and retrieval accesses of each cluster/site for a specific data is accumulated and considered for performing data allocation.

B. Allocation Requirements

Given that there is a set of N disjoint partitions P = {P1, P2, ... , Pn} required by set of K queries Q = {Q1, Q2, ..., Qk}, are to be assigned to a set of M network sites S={S1, S2, …., Sm} which are grouped into Cs clusters Cs = {Cs1, Cs2, …., $C_S$cn} in a fully connected network. Normally, allocation model seeks to find the optimal distribution of each partition (P) over clusters Cs, and consequently on cluster's own sites individually.

C. Allocation Phases

Phase 1 (Scenario 1): each partition would be allocated to all clusters of sites as data replication adopted. This step comes in favour of decreasing transmission costs as well as increasing data locality and availability; specifically when retrieval operations are outnumbered update operations.

Phase 1 (Scenario 2): based on the proved-to-be-effective theory of [1], this scenario is going to be done so that each partition is allocated to cluster of maximum access cost. In other words, Total Access Cost of each sites' Cluster (TACC) is bound to be used as measure of partitions assignment over clusters. This scenario afterwards has recently been shown to be much more effective specifically when update operations are outnumbered retrieval operations [2, 26].

Phase2 for both Scenarios: partitions are to be scattered over sites of each cluster individually so that each partition would be given to one site in each cluster.

For Data Replication, data replication model, which is drawn in [Adel et al, 2017] based on [wise, 2016], is expertly utilized as it has been proven to have huge positive impact on overall DDBS performance. However, this model is slightly modified to have it skilfully complied with proposed work of this paper. It worth noting that X_{ik} points to partition F_i located in cluster C_k or (site S_k), and Y_k indicate to that cluster/site M already in use. Thus, an integer linear program (ILP) to represent this problem presented as follows;

$$Minimize \sum_{k=1}^{m} y_k \qquad 4$$

$$\sum_{i=1}^{N} X_{ik} = 1 \qquad k = 1, \dots, m \qquad 5$$

$$\sum_{k=1}^{m} C_i X_{ik} \leq C_{yk}, \qquad i = 1, \dots, m \qquad 6$$

VI. The Data Replication Model

Data replication model, which is drawn in [2] based on original idea of [24], is set to be substantially utilized. However, this model would be slightly modified to have model completely complied with the proposed technique of this work.

VII. Experimental results

To proof proposed concepts, this work has been properly implemented on the proposed relation "Medicine" (table 2) as per description given in table 1, and this implementation is performed using code in C++ program language with the same data requirements of [1]. In the first step, relation is to be identified to be partitioned; and in the last step all resulted partitions would be appropriately scattered over network sites. According to [1], data requirements of database can be provided explicitly by administrator of DDBSs or could be generated (adopted in this implementation) using a generator for a given attributes predicates and applications over network sites or even computed by tester if necessary. For this implementation, this code is running on processor 3.3 GHz Intel (R) Dual Core(TM) i5CPU. The main memory is 2 GB and hard drive is 250-GB. Just to proof concepts proposed in this work and for the sake of simplicity as well, one single experiment is exclusively conducted with assuming fully-connected networks of four sites (Figure 2, Table 3).

Attributes	Type	Length (Bytes)
M-no	Nominal	4
M-name	Categorical	30
Expir-date	Categorical	40
M-id	Categorical	4
Price	Numerical	3
Manufacture	Categorical	5
Store-id	Nominal	4

Table (1): Medicine database description

M-no	M-Name	Expir-date	M-Id	Price	Manufacturer	Store-Id
1	Aspren	01/02/2020	M12	2000	USA	S2
2	Antiobiotic	01/06/2020	M13	1400	UK	S3
3	Katflam	12/01/2225	M14	1750	Canada	S2
4	Pain killer	22/09/2021	M15	2100	USA	S1
5	Gaze	05/08/2024	M16	1500	USA	S3
6	Carnvita	02/02/2020	M18	1150	Germany	S4
7	Cajova	09/09/2021	M19	1300	Germany	S2
8	Jlocovag	03/01/2224	M20	1200	UK	S1
9	Folic B	07/07/2225	M21	1500	Canada	S1

Table 2: Medicine Relation

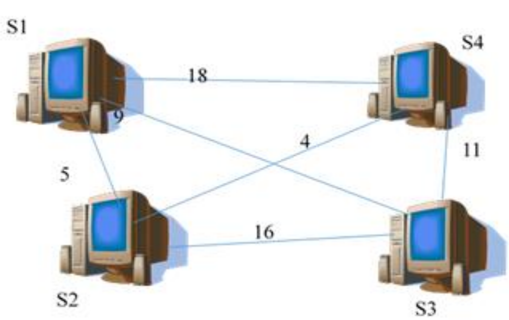

Figure 2: Network Sites (four sites)

Moreover, suppose having communication cost matrix (CSM) between sites (table 3) and sites' constraints (table 4).

Site/Site	S1	S2	S3	S4
S1	0	5	9	18
S2	5	0	16	4
S3	9	16	0	11
S4	18	4	11	0

Table 3: CSM

S #	Capacity (C) in byte	Partition Limit (PL)
S_1	1000	4
S_2	900	1
S_3	250	3
S_4	870	3

Table 4: Network Sites with Constraints (four Sites)

A. Partitioning Process

As mentioned earlier, for partitioning phase, the same procedure presented in [1] is also strictly followed in this extended work which accidentally lead to have the same results of partitioning process (in this example) for both works. To recap, the execution steps have partly shown in following steps (all tables and pictures are taken from real implementation). In step 1; all requirements information of model are to be accurately entered (Figure 3, Table 5).

Enter no of queries : 5
Enter no of sites : 4
Enter no of attributes : 6
For attribute 1, enter no of predicates : 0
For attribute 2, enter no of predicates : 3
For attribute 3, enter no of predicates : 0
For attribute 4, enter no of predicates : 3
For attribute 5, enter no of predicates : 0
For attribute 6, enter no of predicates : 3

Figure 3: The needed Information (four sites)

Site	Query	Frequency	Operation Mod	Birth-date			Salary			Location		
Site	Query	Frequency	RF/UF	P1	P2	P3	P1	P2	P3	P1	P2	P3
S1	Q1	3	RF	1	0	0	1	1	2	0	0	1
S1	Q1		UF	2	0	1	1	0	0	2	2	0
S1	Q2	5	RF	3	1	0	2	3	1	1	2	0
S1	Q2		UF	0	1	1	2	2	0	1	2	0
S2	Q2	2	RF	3	1	0	2	3	1	1	2	0
S2	Q2		UF	0	1	1	2	2	0	1	2	0
S2	Q3	4	RF	0	2	1	2	5	0	1	3	1
S2	Q3		UF	1	0	0	2	1	0	2	0	1
S3	Q1	6	RF	1	0	0	1	1	2	0	0	1
S3	Q1		UF	2	0	1	1	0	0	2	2	0
S3	Q4	8	RF	2	0	2	0	1	1	1	1	0
S3	Q4		UF	0	2	1	1	2	0	1	3	0
S4	Q4	9	RF	2	0	2	0	1	1	1	1	0
S4	Q4		UF	0	2	1	1	2	0	1	3	0
S4	Q5	3	RF	1	0	1	2	0	1	2	2	1
S4	Q5		UF	1	0	2	3	1	0	0	0	3

Table 5: Attribute Retrieval and Update frequency Matrix (ARUM)

Step3; after applying partitioning steps described in section 3, the price attribute is still the candidate partitioning attribute (CA) and predicate (Conditions, C) set is produced, C = {c1:

price > 1500, c2: price < 1500, c3: price = 1500}, and partitions are drawn in tables (6-8).

M-no	M-Name	Expir-date	M-Id	Price	Manufacturer	Store-Id
1	Aspren	01/02/2020	M12	2000	USA	S2
3	Katflam	12/01/2225	M14	1750	Canada	S2
4	Pain killer	22/09/2021	M15	2100	USA	S1

Table 6: First partition

M-no	M-Name	Expir-date	M-Id	Price	Manufacturer	Store-Id
2	Antiobiotic	01/06/2020	M13	1400	UK	S3
6	Carnvita	02/02/2020	M18	1150	Germany	S4
7	Cajova	09/09/2021	M19	1300	Germany	S2
8	Jlocovag	03/01/2224	M20	1200	UK	S1

Table 7: Second partition

M-no	M-Name	Expir-date	M-Id	Price	Manufacturer	Store-Id
5	Gaze	05/08/2024	M16	1500	USA	S3
9	Folic B	07/07/2225	M21	1500	Canada	S1

Table 8: Third partition

B. Allocation Process

Based on allocation model presented in section 5, and the given matrices of ARUM and CSM, data allocation process would be running as follows;

Phase (1): By multiplying ARUM with CCM and piled values based on clusters, TFRUC matrix would be drawn in table (9) to express total of retrieval and update frequency over clusters.

Q# / F#	F1	F2	F3
C1	240	320	70
C2	230	265	155

Table 9: TFRUC

Phase (2), by multiplying ARUM with CSM, TFRS and TFUS matrices would be produced as shown in tables (10-11). These matrices would be used to individually assign partitions into sites of clusters. In other words, TFRS (total of retrieval frequency over sites) and TFUS (total of Update frequency over sites) would be used to determine the precisely-calculated threshold of partitions' allocation over sites (inside each cluster).

Q# / F#	F1	F2	F3
S1	222	418	406
S2	185	350	423
S3	375	677	263
S4	348	582	426

Table 10: TFRS

Q# / F#	F1	F2	F3
S1	510	562	0
S2	361	390	0
S3	507	449	0
S4	436	388	0

Table 11: TFUS

TFRS and TFUS would be used to determine the precisely-calculated threshold of partitions' allocation over sites (inside each cluster).

As per constraints of sites, the allocation process, which is accomplished while site constraints are maintained, for partitions over network sites (for this experiment of four sites) is shown in tables (12-15). Thus, tables (12; 13) shows final partitions' allocation according to [1] and tables (14-15) display final partitions' allocation as per newly-proposed data allocation scenario for [1] and present work.

Partition/Site	S1	S2	S3	S4
F1	1			
F2	1	1	0 (Capacity Violation)	1
F3	1	0(Partition Limit Violation)	1	1

Table 12: Final Partitions' Allocation [1]

Partition/Site	S1	S2	S3	S4
F1			0 capacity violation so to site of next max	1
F2	1		0 capacity violation so to site of next max	
F3				1

Table 13: Final Partitions' Allocation ([1], no replication)

It is worth referring that the scenario drawn in table (13) is newly-proposed scenario in this work. In the sense that [1] just adopted replication scenario which is given in table (12).

Partition/Cluster	C1		C2	
Partition/ Site	S1	S3	S2	S4
F1	1			1
F2	1	0 (capacity violation)	1	1
F3	1	1	0(Partition Limit Violation)	1

Table 14: Final Partitions' Allocation [present work-replication adopted]

Partition/Cluster	C1		C2	
Partition/ Site	S1	S3	S2	S4
F1	1			
F2	1	0(capacity violation)		
F3			1	1

Table 15: Final Partitions' Allocation [present work- no replication adopted]

VIII. CONCLUSIONS

In this work, an extended technique for horizontal partitioning is crucially integrated with newly-proposed clustering algorithm for network sites and mathematically-based cost-effective data allocation, and replication, model. It is worth repeating that this work comes as an extension setup for [1]. This work, like [1], performs partitioning and allocation on the fly that no supplemental complexity is being observed to allocate data partitions over network sites. Additionally, site clustering algorithm is accurately planned so that similar sites (in terms of communication costs) have been clustered together in step ahead of conducting data allocation. Meanwhile, as data allocation is known to have played a significant role in DDBS design and performance alike; in this work, data allocation is fully done using proposed cost-effective model. A different data allocation scenarios are set to be considered that data replication is conducted using proposed replication model. As a result of such precisely-build technique, a significant enhancement has been believed to be recorded in terms of overall DDBSs performance through decreasing transmission costs among the sites of network as distributed query under processing. Finally, experimental results are exclusively illustrated for one single experiment to illustratively demonstrate work's mechanism as well as to primarily meet two goals: to proof concepts of this work, and to show behaviors of both works.

A. Future Work

To further proof proposed concepts and confirms proposed work's superiority, this work would be further run on several datasets of different sizes with diversifying number of queries and sites (in fully-connected network). In other words, putting technique to many different tests under varied circumstances. Moreover, theoretical and internal and external evaluations are going to be extensively made for both works so that all results of all considered problems and their experiments are going to be compared against each other. In the sense that the present work is expected to be accurately evaluated against [1] on the basis of drawn objective function of this work which is originally taken from [1], and significantly amended to reflect on a substantial actual reality for transmission costs.

ACKNOWLEDGMENT

The authors would like to deeply express their sincere appreciation to Prof. Dr. Taha Morsi Elgindy (Mathematics Department, Assuit University) for his valuable support and worthy guidance during this proposal. Additionally, the authors would also take this opportunity to sincerely express their big thanks to reviewers for their valuable comments.

REFERENCES

[1] Hassan I. Abdalla. "A synchronized design technique for efficient data distribution." Computers in Human Behavior. Volume 30, Pp 427–435. (2014).

[2] Adel A. Sewisy, Ali Abdullah Amer, Hassan I. Abdalla. "A Novel Query-Driven Clustering-Based Technique for Vertical Fragmentation and Allocation in Distributed Database Systems." International Journal on Semantic Web and Information Systems (IJSWIS), Volume 13(2). (2017).

[3] S. Ceri, M. Negri, and G. Pelagatti. "Horizontal data partitioning in database design." ACM SIGMOD international conference on Management of data. Pp 128-136. (1982).

[4] [4] S.Ceri, B.Pernici and G. Wiederhold. "Optimization Problems and Solution Methods in the Design of Data Distribution." Journal Information Systems. Volume 14 Issue 3. Pp 261 - 272. (1986).

[5] Yanchun Zhang, Maria E. Orlowska. "On Fragmentation Approaches for Distributed Database Design." Information Sciences – Applications.. Volume 1, Issue 3, Pp 117-132. (1994).

[6] P. Surmsuk and Thanawastien, S. "The integrated strategic information system planning Methodology." Enterprise Distributed Object Computing, 11th IEEE International Conference. (2007).

[7] Ali A. Amer and Hassan I. Abdalla. "Dynamic Horizontal Fragmentation, Replication and Allocation Model In DDBSs." IEEE International Conference on Information Technology and e-Services, Sousse, Tunisia. (2012).

[8] S. Harikumar, R. Ramachandran. "Hybridized fragmentation of very large databases using clustering." IEEE Signal Processing, Informatics, Communication and Energy Systems (SPICES). Pp 1-5. (2015).

[9] Jon Olav Hauglid, Norvald H. Ryeng and Kjetil Norvag. "Dynamic Fragmentation and Replica Management in Distributed Database Systems." Journal of Distributed and Parallel Databases. Vol. 28 No. 3, pp. 1- 25. (2010).

[10] Ahmed E. Abdel Raouf, Nagwa L. Badr and Mohamed Fahmy Tolba, "Distributed Database System (DSS) Design Over a Cloud Environment." © Springer International Publishing AG, Multimedia Forensics and Security. Pp.97-116, (2017).

[11] Xuemin Lin, M. Orlowska ; Yanchun Zhang. "On data allocation with the minimum overall communication costs in distributed database design." Computing and Information, fifth International Conference. (1993).

[12] Yin-Fu Huang, Jyh-Her Chen. "Fragment Allocation in Distributed Database Design." Journal of Information Science and Engineering. (2001).

[13] Leon Tâmbulea.; Manuela. Horvat. "Dynamic Distribution Model in Distributed Database." International Journal of Computers, Communications & Control; Supplement. Vol. 3 Issue 3, Pp 512.515. (2008).

[14] Amita Goyal Chin. "Incremental Data Allocation and Reallocation in Distributed Database Systems." Data warehousing and web engineering, IRM Press Hershey, PA, United States. Pp 137-160. (2002).

[15] Hassan I. Abdalla, Ali A. Amer and Hassan Mathkour. "Performance Optimality Enhancement Algorithm in DDBS (POEA)." Journal of Computers in Human Behavior. 30, 419–426. (2014).

[16] Nilarun Mukherjee. "Synthesis of Non-Replicated Dynamic Fragment Allocation Algorithm in Distributed Database Systems." ACEEE Int. J. on Information Technology. Vol. 01, No. 01. (2011).

[17] Dejan Chandra Gope. "Dynamic Data Allocation Methods in Distributed Database System." American Academic & Scholarly Research Journal. Vol. 4, No. 6. (2012).

[18] Arjan Singh. "Empirical Evaluation of Threshold and Time Constraint Algorithm for Non-replicated Dynamic Data Allocation in Distributed Database Systems. Proceedings of the International Congress on Information and Communication Technology, Advances in Intelligent Systems and Computing 439. (2016).

[19] T. Ulus and M. Uysal. "A Threshold Based Dynamic Data Allocation Algorithm." A Markove Chain Model Approach. Journal of Applied Science. vol. 7, Issue 2, Pp 165-174. (2007).

[20] 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. (2009).

[21] Raju Kumar, Neena Gupta. An Extended Efficient Approach to Dynamic Fragment Allocation in Distributed Database Systems, I J C T A. Pp. 473-482 © International Science Press. (2016).

[22] Wiese, L. “Horizontal fragmentation and replication for multiple relaxation attributes.” Data Science (30th British International Conference on Databases). Pp. 157-169. Springer. (2015).

[23] L. Wiese, T. Waage and F. Bollwein. “A Replication Scheme for Multiple Fragmentations with Overlapping Fragments.” The Computer Journal. (2016).

[24] Rizik M.H. Al-Sayyed, Fawaz A. Al Zaghoul, Dima Suleiman, Mariam Itriq, Ismail Hababeh. “A new Approach for Database Fragmentation and Allocation to Improve the Distributed Database Management System Performance.” Journal of Software Engineering and ApplicationV7 . 891-905. (2014).

[25] M. Tamer Ozsu and Patrick Valduriez. Principles of Distributed Database Systems. 3Edition, New Jersey: Prentice-Hall. (2011).

[26] Bellatreche, L. and Kerkad. “A. Query interaction based approach for horizontal data partitioning.” International Journal of Data Warehousing and Mining, (IJDWM). Volume 11, Pp44-61. (2015).