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

ALI A. AMER
Computer Science Department
Taiz University
YEMEN
aliaaa2004@yahoo.com

ADEL A. SEWISY
Computer Science Department
Assiut University
EGYPT
adel.mohamed@fci.au.edu.eg

Abstract: - In this work, an extended heuristic horizontal partitioning, and data allocation approach is critically evolved. As a matter of fact, the key focus behind this work is to introduce an efficient best-fitting solution in purpose of boosting DDBS rendering through presenting an intelligent data partitioning and allocation approach. However, as partitioning technique is already developed, this work aims at extending this technique by having it skilfully integrated with hierarchically-inspired site clustering algorithm and mathematical model for data allocation, including data replication, for the sake of producing an effective approach. Consequently, this approach is set to be promising and capable of tremendously lessening the overall cost of data transmission (TC). It is believed that such significant extension is to overwhelmingly be a potential progress of profoundly beneficial effects on overall DDBS performance.

Key-Words: - : Horizontal; Partitioning; Allocation; Replication; Site Clustering; DDBS; Optimization

1 Introduction

Generally speaking, 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 paper 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 clustering of sites adoption would come with remarkable benefits in terms of TC reduction as shown in [2]. Moreover, mathematical cost model is given in the sake of paving the way to find much more efficacious data allocation (and replication) model. In short, contributions of this work are clearly listed as follows;

1. The objective function drawn in [1] is critically amended so that transmission cost (including query costs) is significantly reflected.
2. For network site clustering, a hierarchical based algorithm is adopted.
3. Mathematically drawing data allocation 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 completely contain work's modifications including sites grouping.
4. Unlike [1], data replication model is effectively involved.
5. Presenting illustrative step-by-step demonstration with experimental results of one single experiment for both works ([1] and present work) in a clear way to show their behaviours as well as proof proposed concepts.

The remaining of this paper is planned as follows; section (2) explore earlier studies which are closely related to this work. In section 3, technique's methodology, including architecture, is briefly deliberated. Algorithm of site clustering is stated in section 4. In section 5, proposed data allocation, including replication, model is clearly presented. In section 6, to proof concepts of this work, a hypothetical experimental results for one single

experiment are vividly drawn. Lastly, conclusions and future work directions are given in section 7.

2 Related work

As a matter of fact, considerable number of Horizontal Partitioning (HP) methods have been proposed in literature in consecutive steps just to enhance 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 should meet 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 needed 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 that the closely correlated data were grouped together. Experimental results demonstrated that this clustering-based method were better in diminishing access time. Meanwhile, to maximize data locality, [9] proposed a decentralized approach for dynamic table fragmentation and allocation in DDBS (DYFRAM) based on recorded access history. 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] proposed an enhanced 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 addressed in [11] aiming at lessening the overall communication cost so that two algorithm were developed. By the same token, a model to draw queries behavior in DDBS was presented in [12]. Two heuristic algorithms were given to find a near-optimal allocation scenario in terms of reducing communication costs. Compared to [11], this algorithms was shown to be close enough from being an optimal. Meanwhile, [13] presented dynamic data allocation method aiming at lessening transmission cost, considering database catalog as the only storing place for required data as method implemented. As a new of its kind, [14] sought to give partial data reallocation and full reallocation heuristics to minimize costs and maintain complexity under control. 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 algorithms [20]. This work was shown to be more effective in terms of long-term performance than threshold algorithms as access frequency pattern changes rapidly. However, [21] gave an extended allocation approach 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 ensuring that overlapping is being eliminated. 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]. A novel soft data locality constraints based on partitions' affinity was developed, and DRP problem was then re-formalized as an integer linear program. Data insertion and deletion were considered and runtime performance was analyzed as well.

3 Proposed Approach

In this research, for partitioning phase, all requirements, heuristics, definitions, notations, and

formulas drawn in [1] are all strictly used. The extension part however is being carefully made through incorporating both site clustering algorithm and mathematically-designed data allocation and replication models (Figure 1). Nevertheless, objective function of [1] is majorly amended as presented in equations (1-3).

3.1. Objective Function

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

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

$$TC_{total} = TC_R + TC_U \quad (3)$$

Where CMS_{ij} is expressed as follows;

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

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

Finally, the **step eight** of heuristics drawn in [1] is also modified and drawn in section (5).

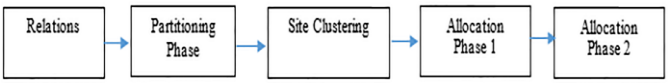

Fig1: The Proposed System Phases

4 Site Clustering

The presented algorithm of site clustering has been made based on concept of hierarchical clustering, especially as initial clusters are to be formed. Then, this method is to be entirely kept proceeding based on the least average of communication cost between sites to decide site's belonging as site being considered 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 chiefly in the non-replication scenario [2].

In the meantime, the symmetry average of communication cost 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 and cost between the same sites is considered to be a zero or, table 2. The same presumption goes to cluster matrix, as shown in table (1). It is worth referring that the costs between clusters are set to be the shortest path between their closest points. Meanwhile, the costs between points in the same clusters are calculated as the average costs of them all [2]. For this work, table (1) exhibits communication costs matrix between sites under consideration.

Site/Site	S1	S2	S3	S4	S5	S6
S1	0	10	8	2	4	6
S2	10	0	7	3	5	4
S3	8	7	0	3	2	5
S4	2	3	3	0	11	5
S5	4	5	2	11	0	5
S6	6	4	5	5	5	0

Table 1: Communication Costs between Sites

5 Proposed Allocation Model

5.1 Problem Formulation

In DDBS, it has been taken for granted that the optimal solution to promote DDBSs performance is to properly partition data, and carefully allocate data 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.

5.2 Allocation Requirements

Given that there is a set of N disjoint partitions P = {P_1, P_2, ... , P_n} required by set of K queries Q = {Q_1, Q_2, ..., Q_k}, are to be assigned to a set of M network sites S={S_1, S_2, …., S_m} which are grouped into Cs clusters Cs = {Cs_1, Cs_2, …., CS_{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. Thus, the allocation problem can be mathematically expressed by a function from the set of partitions to the set of clusters of sites, equation (4).

$$Partitions \xrightarrow{distributed\ over} Clusters\ (Cs) \xrightarrow{assigned\ to} Sites\ (S) \quad (4)$$

5.3 Allocation Scenarios

5.3.1 Scenario 1: Phase 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.

5.3.2 Scenario 2: Phase 1;

Based on the proved-to-be-effective theory of [1], this scenario is 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 is recently shown to be much more effective specifically when update operations are outnumbered retrieval operations [2, 26].

5.3.3 Phase2 for both Scenarios 1 and 2:

Partitions are to be scattered over sites of each cluster individually to place them into sites. Firstly, a threshold would be tacitly calculated based on Average of Update Cost (AUC) and Average of Retrieval Cost (ARC) of each partition. Therefore, whenever P's AUC is greater than P's ARC, the triggered partition would be assigned to site of maximum update cost inside its relative cluster providing that cluster/site's constraints have never been violated. However, 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 than AUC, that partition is to be allocated to all sites requesting it as it is being done in [1]. Consequently, the ideal case is set to be satisfied and DDBSs' response time, disk access and overall performance are bound to have got reinforced.

5.4 Allocation Costs Function

$$TSFRP = \sum_{l=1}^{m}\sum_{i=1}^{f}\sum_{j=1}^{m}\sum_{k=1}^{q} QFM_{ik} * RFM_{ki} * CMS_{lj} \quad (5)$$

$$TSFUP = \sum_{l=1}^{m}\sum_{i=1}^{f}\sum_{j=1}^{m}\sum_{k=1}^{q} QFM_{ik} * UFM_{ki} * CMS_{lj} \quad (6)$$

$$TFRUP = \sum_{i=1}^{m}\sum_{j=1}^{f} TSFRM_{ij} + TSFUM_{ij} \quad (7)$$

$$TCSFRP = \sum_{l=1}^{Cn}\sum_{i=1}^{f}\sum_{j=1}^{m}\sum_{k=1}^{q} QFM_{ik} * RFM_{ki} * CMC_{lj} \quad (8)$$

$$TCSFUP = \sum_{l=1}^{Cn}\sum_{i=1}^{f}\sum_{j=1}^{m}\sum_{k=1}^{q} QFM_{ik} * UFM_{ki} * CMC_{lj} \quad (9)$$

Regarding data replication, data replication model, which is drawn in [2] based on original idea of [24], is expertly used. However, this model is slightly modified to capable it of complying with proposed work of this paper. Thus, an integer linear program (ILP) to represent this problem presented as follows;

$$Minimize \sum_{k=1}^{m} y_k \quad (10)$$

$$\sum_{i=1}^{N} X_{ik} = 1 \qquad k = 1, \dots, m \quad (11)$$

$$\sum_{k=1}^{m} C_i X_{ik} \le C_{yk}, \qquad i = 1, \dots, m \quad (12)$$

$X_{ik} \in \{0,1\}$ $\quad k = 1, \dots, m ; i = 1, \dots , n$ (13)

$Y_{ik} \in \{0,1\}$ $\quad k = 1, \dots, m ; i = 1, \dots , n$ (14)

Finally, Table (2) describes two constraints of sites, represented in virtual capacity (in byte), partitions limit allowed to be placed at each site.

Site	Capacity (C) in byte	Partition Limit (PL)
S_1	1000	5
S_2	900	1
S_3	250	3
S_4	870	3
S5	950	2
S6	710	2

Table 2: Network Sites with Constraints

6 Experimental Results

This work has been properly implemented on the same relation "Staff", table 4, as per description given in table 3. For this simple implementation, C++ code is running on processor 3.3 GHz Intel (R) Dual Core(TM) i5CPU, main memory of 2 GB and hard drive of 250-GB. It is of major importance to say that because of space limitation of this paper and to only proof concepts proposed as well as for the sake of simplicity, one single experiment is particularly conducted with assuming a fully-connected network of six sites.

Attributes	Type	Length (Bytes)
Staff-no	Nominal	3
Staff-name	Categorical	33
Hire-date	Categorical	30
Pay	Numerical	4
Dept	Categorical	7
Course-id	Nominal	3

Table (3): Employee dataset

Staff-no	Staff-name	Hire-date	Pay	Dept	Course-Id
1	Anna	05/03/2012	10000	CS	22
2	Browni	02/02/2011	11000	IS	31
3	Swayer	05/03/2012	7050	ES	22
4	Malik	12/12/2011	12000	ES	11
5	Susan	03/03/2013	6500	ES	31
6	Jasmin	04/02/2013	6500	IS	14
7	Jessica	06/04/2012	7500	CS	22
8	Jouvani	07/03/2011	12000	CS	11
9	Salem	02/03/2012	10000	IS	31

Table 4: Employee Relation

As mentioned earlier, for partitioning phase, the same procedure presented in [1] is also strictly followed in this work. As a result, the same outcomes of partitioning process are obtained (in this experiment) for both works. To recap, the execution steps have partly illustrated in following steps (all tables and pictures are taken from real implementation). In step 1; all information requirements of model are accurately given (Figure 2).

```
Enter no of queries                          : 5
Enter no of sites                            : 6
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
```

Fig2: The needed Information

Step 2; based on these requirements along with proposed cost model, ARUM matrix is set to entirely be taken from [1] **for the first four sites**, while information of newly-added sites S5 and S6 is drawn in table 5.

S #	Q #	Frequency	Mod	Birth-date			Salary			Location		
			RF/ UF	P 1	P 2	P 3	P 1	P 2	P 3	P 1	P 2	P 3
S 5	Q 1	2	RF	1	0	0	1	1	2	0	0	1
S 5	Q 1		UF	2	0	1	1	0	0	2	2	0
S 5	Q 3	3	RF	0	2	1	2	5	0	1	3	1
S 5	Q 3		UF	1	0	0	2	1	0	2	0	1
S 5	Q 5	2	RF	1	0	1	2	0	1	2	2	1
S 5	Q 5		UF	1	0	2	3	1	0	0	0	3
S 6	Q 2	2	RF	3	1	0	2	3	1	1	2	0
S 6	Q 2		UF	0	1	1	2	2	0	1	2	0
S 6	Q 3	1	RF	0	2	1	2	5	0	1	3	1
S 6	Q 3		UF	1	0	0	2	1	0	2	0	1

Table 5: ARUM

Step3; after applying cost model, Pay attribute is selected as the candidate partitioning attribute (CA). With assuming that Predicate Set of Pay is given as follows; PS = {PS1: Pay > 10000, PS2: Pay < 10000, PS3: Pay = 10000}; then, partitions are set to be drawn as shown in tables (6-8).

Staff-no	Staff-name	Hire-date	Pay	Dept	Course-Id
2	Browni	02/02/2011	11000	IS	31
4	Malik	12/12/2011	12000	ES	11
8	Jouvani	07/03/2011	12000	CS	11

Table 6: First partition

Staff-no	Staff-name	Hire-date	Pay	Dept	Course-Id
3	Swayer	05/03/2012	7050	ES	22
5	Susan	03/03/2013	6500	ES	31
6	Jasmin	04/02/2013	6500	IS	14
7	Jessica	06/04/2012	7500	CS	22

Table 7: Second partition

Staff-no	Staff-name	Hire-date	Pay	Dept	Course-Id
1	Anna	05/03/2012	10000	CS	22
9	Salem	02/03/2012	10000	IS	31

Table 8: Third partition

6.1.Partitions Allocation

As per allocation cost model of this work, the allocation process would be completed in two scenarios each of which is of two phases. Thus, from ARUM matrix along with using the cost functions of section 5, matrices below (9-13) are extracted as follows; (SFRP and SFUP stand for both Frequency Matrices of Partitions' Retrieval and Update over sites).

S#/Q#	Q1	Q2	Q3	Q4	Q5
S1	3	5	0	0	0
S2	0	2	4	0	0
S3	6	0	0	8	0
S4	0	0	0	9	3
S5	2	0	3	0	2
S6	0	2	1	0	0

Table 9: OFM

S#/ P#	P1	P2	P3
S1	13	18	11
S2	12	26	2
S3	6	14	18
S4	6	9	12
S5	12	17	6
S6	6	11	2

Table 10: SFRP

S#/ P#	P1	P2	P3
S1	462	442	442
S2	472	232	313
S3	463	224	163
S4	424	393	143
S5	443	332	434
S6	426	214	442

Table 11: TFPRS

S#/ P#	P1	P2	P3
S1	13	10	0
S2	12	8	0
S3	14	16	0
S4	18	21	0
S5	14	5	0
S6	6	5	0

Table 12: SFUP

S#/ P#	P1	P2	P3
S1	363	333	3
S2	376	343	3
S3	333	432	3
S4	288	172	0
S5	368	368	0
S6	356	302	0

Table 13: TFPUS

TRFM and TFUM would be used to determine the precisely-calculated threshold of partitions' allocation over sites as presented in [1]. Meanwhile,

the next matrices (14-17) are drawn as a result of implementing allocation cost model of section (5). (CFRP and CFUP stand for both Frequency Matrices of Partitions' Retrieval and Update over Clusters of Sites)

S#/ P#	P1	P2	P3
S1	624	824	224
S2	650	754	310
S3	560	676	160
S4	530	570	158
S5	588	752	232
S6	602	714	254

Table 14: TFRUP

C#/ P#	P1	P2	P3
C1	18	37	0
C2	19	27	23
C3	18	31	24

Table 15: CFRP

C#/ P#	P1	P2	P3
C1	18	13	0
C2	31	31	0
C3	28	21	0

Table 16: CFUP

C#/ P#	P1	P2	P3
C1	380	434	189
C2	246	306	84
C3	330	424	89

Table 17: TCSFRUP

As per constraints of sites, the allocation process for partitions over sites is shown in tables (18 - 21). Therefore, tables (20; 21) show final partitions' allocation for partitions according to [1], and tables (22; 23) display final partitions' allocation of present work. It is worth indicating that allocation is just accomplished while site constraints are kept maintained.

P#/S#	S1	S2	S3	S4	S5	S6
P1	0	1	0 capacity violation	0	0	0
P2	1	0 partition limit violation	0 capacity violation	1	1	1
P3	1	0 partition limit violation	1	1	1	1

Table 18: Final Partitions Allocation ([1], replication adopted)

P# /S#	S1	S2	S3	S4	S5	S6
P1		1				
P2	1					
P3	1	0 partition limit violation				

Table 19: Final Partitions Allocation ([1], no replication)

P#/C#	C1		C2		C3	
P# / S#	S2	S6	S1	S4	S3	S5
P1	1			1	0 (capacity violation)	1
P2	0 (partition limit violation)	1		1	0 (capacity violation)	1
P3	0 (partition limit violation)	1	1	1	1	1

Table 20: Final Partitions Allocation (present work- replication adopted)

P#/C#	C1		C2		C3	
P#/ S#	S2	S6	S1	S4	S3	S5
P1	1					
P2	0 (partition limit violation)	1				
P3	0 (partition limit violation)	1				

Table 21: Final Partitions Allocation (present work- no replication adopted)

7 Conclusion and Future Work

In this work, an extended approach for horizontal partitioning is suggested and crucially integrated with 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 previous work [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) are to be clustered together in step ahead of conducting data allocation. Meanwhile, data allocation is known to have played a significant role in DDBS design and performance alike. In this work, therefore, it is fully done using proposed cost-effective model. A different data allocation scenarios are being considered that data replication is conducted using proposed replication model. A threshold of retrieval and update costs has been used to decide whether or not replicating partitions over sites. As a result of such precise data placement procedure, a significant enhancement has been believed to be recorded in terms of overall DDBSs performance through decreasing transmission costs among the sites of network. This undeniable fact however is going to be strongly proved in follow-up work with presently-given objective function being in mind. Constraints of clusters and sites are also considered to stimulate the real-world DDBS as well as strengthen the proposed work efficiency. Finally, due to the limited space of this work, experimental results (for one single experiment) are exclusively done 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.

7.1 Future Work

The follow-up work is completely set to be directed toward conducting more experiments on several real datasets of different sizes with diversifying number of queries and network sites to get on with many tests under different circumstances. Moreover, theoretical and internal and external evaluations are going to be extensively made along with comparing all results of all problems and their experiments under consideration. 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 substantial actual reality of transmission costs. In short, all these suggestions would be effectively addressed in the follow-up work which set to come in purpose of theoretically and experimentally demonstrating extended work's superiority and effectiveness.

Acknowledgement

The authors would like to deeply express their sincere appreciation to Prof. Dr. Taha Morsi Elgindy (Mathematics Department, Assiut University) for his valuable support and worthy guidance during this research. 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, (2014) Pp 427–435. http://www.sciencedirect.com/science/article/pii/S0747563213001374

[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) http://www.igi-global.com/article/a-novel-query-driven-clustering-based-technique-for-vertical-fragmentation-and-allocation-in-distributed-database-systems/176732.

[3] S. Ceri, M. Negri, and G. Pelagatti, Horizontal data partitioning in database design. ACM SIGMOD international conference on Management of data. (1982). Pp 128-136. DOI: http://dl.acm.org/citation.cfm?id=582376

[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. (1986) Pp 261 - 272. http://dl.acm.org/citation.cfm?id=71879.

[5] Yanchun Zhang, Maria E. Orlowska, On Fragmentation Approaches for Distributed Database Design. Information Sciences – Applications.. Volume 1, Issue 3, (1994) Pp 117-132. http://www.sciencedirect.com/science/article/pii/1069011594900051

[6] P. Surmsuk and Thanawastien, S, The integrated strategic information system planning Methodology. Enterprise Distributed Object Computing, 11th IEEE International Conference. (2007). DOI:10.1109/EDOC.2007.48

[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 (ICITeS'2012), Sousse, Tunisia. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6216603&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6216603

[8] S. Harikumar, R. Ramachandran, Hybridized fragmentation of very large databases using clustering. IEEE Signal Processing, Informatics, Communication and Energy Systems (SPICES). (2015) Pp 1-5. DOI: 10.1109/SPICES.2015.7091488

[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, (2010) pp. 1- 25.

[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. (2017). Pp.97-116, DOI: 10.1007/978-3-319-44270-9_5.

[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). http://www.cse.unsw.edu/~lxue/paper/icci93.pdf

[12] Yin-Fu Huang, Jyh-Her Chen, Fragment Allocation in Distributed Database Design. Journal of Information Science and Engineering. (2001). http://www.iis.sinica.edu.tw/page/jise/2001/200105_08.pdf

[13] Leon Tâmbulea.; Manuela. Horvat, Dynamic Distribution Model in Distributed Database. International Journal of Computers, Communications & Control; Supplement. Vol. 3 Issue 3, (2008) Pp 512.515. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.452.770&rep=rep1&type=pdf

[14] Amita Goyal Chin, Incremental Data Allocation and Reallocation in Distributed Database Systems. Data warehousing and web engineering, IRM Press Hershey, PA, United States. (2002) Pp 137-160. http://dl.acm.org/citation.cfm?id=779519

[15] Hassan I. Abdalla, Ali A. Amer and Hassan Mathkour, Performance Optimality Enhancement Algorithm in DDBS (POEA). Journal of Computers in Human Behavior. 30, (2014) 419–426. http://www.sciencedirect.com/science/article/pii/S0747563213001386

[16] Nilarun Mukherjee, Synthesis of Non-Replicated Dynamic Fragment Allocation Algorithm in Distributed Database Systems. ACEEE Int. J. on Information Technology. Vol. 01, (2011) No. 01. http://searchdl.org/public/journals/2011/IJIT/1/1/98.pdf

[17] Dejan Chandra Gope, Dynamic Data Allocation Methods in Distributed Database System. American Academic & Scholarly Research Journal. Vol. 4, (2012) No.6. http://naturalspublishing.com/files/published/15j7d2xw82j2v4.pdf

[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). DOI 10.1007/978-981-10-0755-2_15

[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, (2007) Pp 165-174. http://adsabs.harvard.edu/abs/2007JApSc...7..165U

[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 (2009) No.9. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.512.2367&rep=rep1&type=pdf

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

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

[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, . (2014) 891-905.

[25] M. Tamer Ozsu and Patrick Valduriez, Principles of Distributed Database Systems. (2011). 3Edition, New Jersey: Prentice-Hall. http://www.springer.com/us/book/9781441988331

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