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: QFM

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	264	524	224
S2	274	434	310
S3	260	442	160
S4	242	398	158
S5	220	384	232
S6	246	412	254

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	360	300	0
S2	376	320	0
S3	300	234	0
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, Assuit 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.