Empirical Evaluation of Threshold, Time, Volume, and Distance Constraint Algorithm for Redundant and Non-Redundant (TTVDCA-RNR) Dynamic Data Allocation in Distributed Database Systems

Raju Kumar
Department of Computer Science, Gurukula Kangri Vishwavidyalaya, Haridwar India
rajuk12@gmail.com

Neena Gupta
Department of Computer Science, Kanya Gurukula Mahavidyalaya Dehradun
(Second Campus of Gurukula Kangri Vishwavidyalaya, Haridwar), India
neena71@hotmail.com

***Abstract*—The key idea of this paper is to empirically evaluate the performance of the algorithm TTVDCA-RNR for dynamic data allocation in distributed database system (DDS). Distributed database system performance depends heavily on the allocation of data fragments at different sites. For improving the performance of distributed database system, this algorithm is working on the concept of dynamically placing the data at site(s) where it is most desirable. To study the efficiency of the algorithm TTVDCA-RNR over other similar algorithm, a sample distributed database is used. Comparative results prove that TTVDCA-RNR is more efficient algorithm and significantly improves the overall performance of the system.**

Keywords*- Distributed database system, Redundant, Allocation, Elapsed time, TTVDCA-RNR

I. INTRODUCTION

In the age of globalization and digitization, majority of the organizations are distributed in nature. They have several branches at several locations across the globe. Managing huge amount of data of such organizations from one centralized location is difficult and inefficient. This caused the huge demand for having distributed database, in which data is fragmented and allocated over different sites. Each site can perform its local applications autonomously but must participate in the execution of at least one global application, which requires accessing of data stored at several sites through a communication subsystem [1]. The performance of distributed database system highly depends on the allocation of data fragments at different sites [2]. Allocation of fragments can be performed in redundant and non-redundant manner [3]. For better performance of the distributed database system, initially fragment is allocated to the site(s) which need it most. These initial data allocations may not be beneficial after some time due to changing data access probability of the sites to the fragment. Therefore, an efficient dynamic data allocation mechanism is required.

Several works have been published during past few years, on the issue of dynamic data allocation to the sites of distributed database systems. For non-replicated distributed database [4] presented a threshold algorithm, which reallocate fragments as per changing data access patterns. Further with the addition of time constraint [5] introduced a Threshold and Time Constraint Algorithm (TTCA) for non-redundant dynamic data allocation in distributed database system and [6] presented empirical evaluation of TTCA. Thereafter [7] found that TTCA suffered from scaling problem and proposed an Extended Threshold Algorithm (ETA) which not only eliminated the scaling problem of TTCA but also reduced space requirements. By considering the access threshold, time constraints of database accesses and volume of data transmission [8] introduced a new dynamic fragment allocation algorithm. In this algorithm there was no provision to deal the situation where multiple sites qualify for fragment relocation. To handle this issue [9] extended the work done by [8] by introducing additional parameter of distance and presented an extended algorithm. For dynamic data allocation in replicated and non-replicated distributed database system, [10] presented a systematic survey of 31 research papers. Further by extending the work done by [9], and [8], an efficient algorithm named TTVDCA-RNR was proposed by [11], which was also capable to reallocate data dynamically in redundant distributed database system. Moreover [12] introduced a Clustered Approach to Dynamic Data Allocation (CADDA) algorithm, which dynamically reallocates data fragments to clusters in redundant and non-redundant distributed database system and also reduced the remote data accesses and network overhead. Over cloud environment [13] presented an data allocation and replication technique for distributed database, which can be applied at the initial stage of distributed database design where no information about the query execution is available. For vertical fragmentation and allocation in distributed database systems, [14] introduced a novel query-driven clustering-based technique, which can be applicable to static as well as dynamic distributed database environment.

In this paper, an empirical evaluation of algorithm TTVDCA-RNR for dynamic data allocation in distributed database system is presented and compared with algorithm proposed by [8]. Comparative results demonstrate that TTVDCA-RNR is more efficient algorithm and significantly improves the overall performance of the system. The rest of the paper is organized as follows: In section II a sample distributed database system performance evaluation environment is described. Section III demonstrates algorithms performance and comparison. Finally, the contribution of the study is concluded in section IV.

II. DDS PERFORMANCE EVALUATION ENVIRONMENT

To evaluate the performance of algorithms proposed by [11] and [8], a fully connected sample non-redundant and redundant distributed database system (DDS) is created consisting of 4 sites and 6 fragments as shown in figure 1 and figure 2:

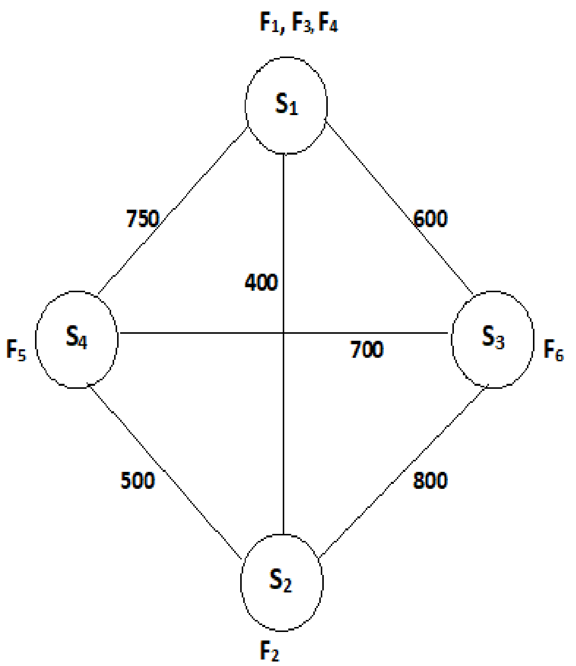

Figure 1. Non-redundant DDS

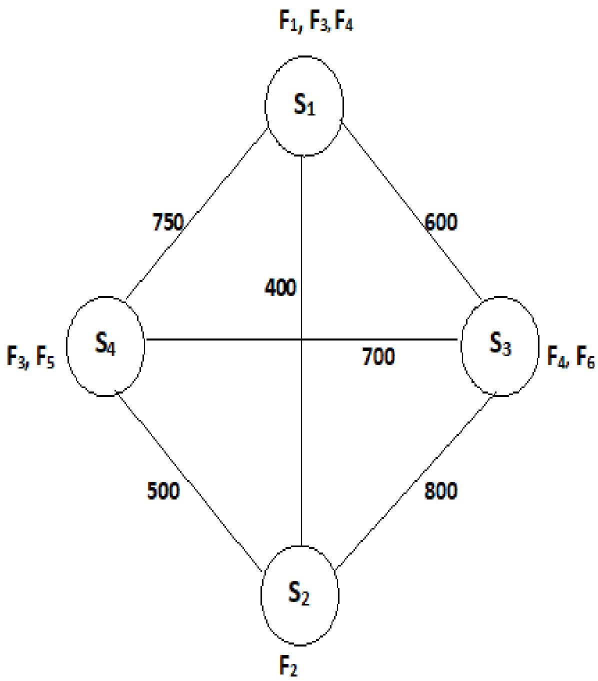

Figure 2. Redundant DDS

To make this paper concise and to the point, details about algorithms proposed by [11] and [8] are not discussed here, but can be found in respective papers. Notations used in this paper and their meaning is shown in table I.

TABLE I. NOTATION WITH MEANING

Notation	Meaning
F_i	The i^{th} data fragment
S_j	The j^{th} site
α	Access threshold for fragment relocation
β	Time constraint for fragment relocation
D_i^j	Distance between site S_i and site S_j

The allocation of fragments over different sites for non-redundant and redundant distributed database is shown in table II and table III.

TABLE II. FRAGMENT ALLOCATION MATRIX FOR NON-REDUNDANT DDS

Site → Fragment ↓	**S_1**	**S_2**	**S_3**	**S_4**
F_1	1	0	0	0
F_2	0	1	0	0
F_3	1	0	0	0
F_4	1	0	0	0
F_5	0	0	0	1
F_6	0	0	1	0

TABLE III. FRAGMENT ALLOCATION MATRIX FOR REDUNDANT DDS

Site → Fragment ↓	**S_1**	**S_2**	**S_3**	**S_4**
F_1	1	0	0	0
F_2	0	1	0	0
F_3	1	0	0	1
F_4	1	0	1	0
F_5	0	0	0	1
F_6	0	0	1	0

$$where \begin{cases} 1, fragment\ F_i\ is\ allocated\ to\ S_j \\ 0, otherwise \end{cases}$$

The distance between different sites is shown in table IV. The distance is shown in km. Each site stores only its respective row of Site Distance Matrix. Obviously $D_i^j = 0$ for all i = j.

TABLE IV. SITE DISTANCE MATRIX

Site	**S_1**	**S_2**	**S_3**	**S_4**
S_1	0	400	600	750
S_2	400	0	800	500
S_3	600	800	0	700
S_4	750	500	700	0

Each site of distributed database system stores the following information:

- At least one fragment, which is allocated on it
- Fragment allocation matrix – to prepare its access plan
- Its respective row of site distance matrix – to know its distance from other sites
- Access Threshold for fragment reallocation (α)
- Time constraint for fragment reallocation (β)
- Its Access_Log table – to store access log record for each access to the fragments allocated to that site

The Access_Log table has following schema –

Access_Log (AFID, ASID, ADateTime, DataVol)

Where attribute AFID represents ID of the fragment which is accessed, ASID represents ID of the site which accesses the fragment, ADateTime represents date and time of fragment access, and DataVol represents volume of data transmitted to and from the accessed fragment.

III. PERFORMANCE EVALUATION AND COMPARISON

In order to evaluate the performance of algorithms two procedures are developed – one as per algorithm proposed by [11] and another as per algorithm proposed by [8]. It is assumed that site S1 accessing its all fragments, and all other remote sites accessing all those fragments of site S1 which are not locally available to those sites. For the sake of simplicity, it is assumed that each site accessing all the fragments of site S1 up to the value of access threshold (α) within given time constraint (β). The records of Access_log table at site S1 are taken in such a way that average volume of data transferred between fragment F1 and site S4, is greater than, the average volume of data transferred between fragment F1 and all other sites including site S1. Thereafter, the two procedures are executed separately at site S1, when site S4 accessing the fragment F1 of site S1. The performance of each algorithm is recorded in elapsed time (in seconds) at constant time constraint (β), different access threshold value (α) and different number of records in Access_Log table. The elapsed time varies for the same query or procedure, when it executed several number of times. Therefore, average elapsed time is taken for comparison. For non-redundant DDS the algorithms performance in elapsed time with respect to time constraint constant (β) of 10 days, and different values of access threshold (α) and different number of records in Access_Log is shown in table V.

TABLE V. ALGORITHMS PERFORMANCE IN NON-REDUNDANT DDS

Value of Access Threshold (α)	No. of Records in Access_Log	Elapsed Time (in second)	
		Algorithm TTVDCA-RNR	Algorithm in paper [8]
1600	19200	0.014	0.017
3200	38400	0.015	0.028
6400	76800	0.022	0.048
12800	153600	0.036	0.083
25600	307200	0.054	0.147

The comparative performance of the two algorithms in non-redundant DDS is shown in figure 3.

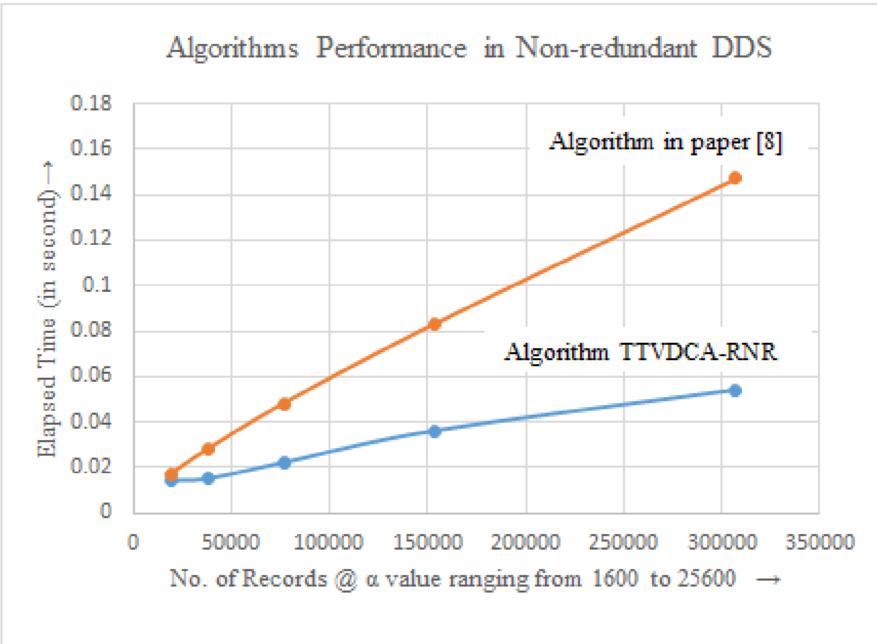

Figure 3: Algorithms performance in non-redundant DDS

Although the algorithm proposed by [8] is for non-redundant DDS, if same is applied for redundant DDS, then for redundant DDS the algorithms performance in elapsed time with respect to time constraint constant (β) of 10 days, and different values of access threshold (α) and different number of records in Access_Log is shown in table VI.

TABLE VI. ALGORITHMS PERFORMANCE IN REDUNDANT DDS

Value of Access Threshold (α)	No. of Records in Access_Log	Elapsed Time (in second)	
		Algorithm TTVDCA-RNR	Algorithm in paper [8]
1600	16000	0.013	0.016
3200	32000	0.014	0.026
6400	64000	0.018	0.042
12800	128000	0.029	0.073
25600	256000	0.046	0.145

The comparative performance of the two algorithms in redundant DDS is shown in figure 4.

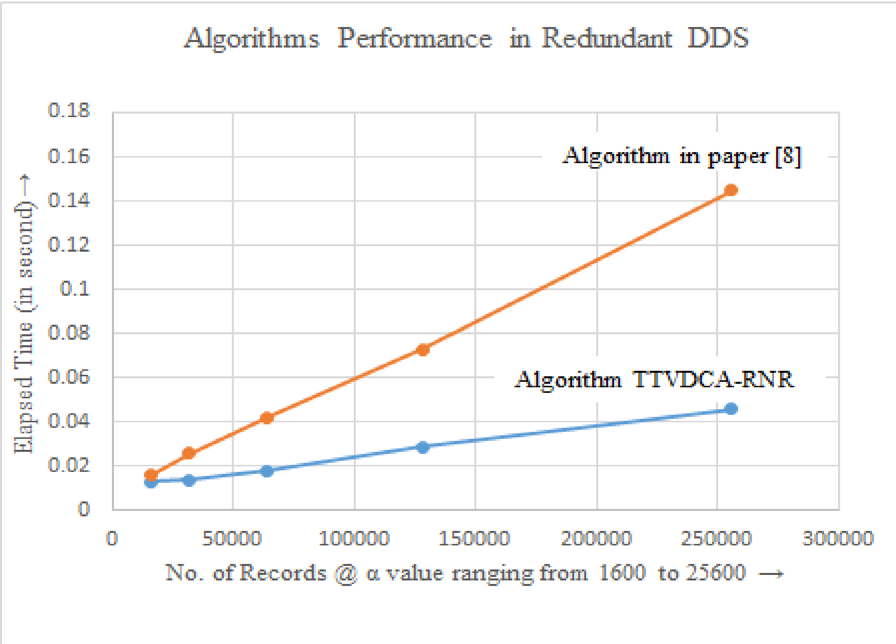

Figure 4: Algorithms performance in redundant DDS

From table V & table VI and figure 3 & figure 4, it is clear that algorithm TTVDCA-RNR is faster than the algorithm proposed by [8]. For the sake of simplicity, only four sites and six fragments are taken into consideration. If the number of sites and fragments are increased, then similar performance results can be found at lower values of access threshold (α). Time taken by algorithm TTVDCA-RNR to delete records older than time constraint (β) is almost equal to negligible, as it has to run a simple query only one time per day. In order to provide equal opportunity to both the algorithms, all records taken are older up to time constraint (β). While in actual, algorithm proposed by [8] have some records older than time constraint (β), which means it has to process more number of records to get the records older up to time constraint (β) as a result it will further take more time as compared to TTVDCA-RNR algorithm. Furthermore, algorithm proposed in [8] allows Access_Log table to store records even older than the time constraint (β), while algorithm TTVDCA-RNR allows Access_Log table to store only records older up to time constraint (β), as a consequence algorithm TTVDCA-RNR is more efficient with respect to storage requirements also. Moreover, algorithm proposed in [8] is unable to handle the situation, when multiple sites qualify for fragment relocation, whereas algorithm TTVDCA-RNR is capable to handle such situation. In such cases algorithm TTVDCA-RNR suggests to reallocate the fragment to the site which is at more distance from the accessing fragment site. Additionally, to reduce the occurrence of simultaneous access, value for column ADateTime (fragment accessing date and time) in the Access_Log table can be stored up to milliseconds/microseconds.

In view of the above results it clearly proves that algorithm TTVDCA-RNR is more efficient than algorithm proposed by [8] and improves the overall performance of the system.

IV. CONCLUSION

Globally the distributed database system is in huge demand. Efficient dynamic data allocation is the key to distributed database performance. In this paper, an idea to empirically evaluate the performance of the algorithm TTVDCA-RNR is presented. Empirical results and explanations prove that the algorithm TTVDCA-RNR is more capable and efficient than other similar algorithm and consequently improves the overall performance of the distributed database system. As a future work, soft computing techniques can be explored to further improve the performance of distributed database system.

REFERENCES

[1] S. Ceri, and G.Pelagatti, Distributed Databases: Principles &Systems, edn. 2008. McGraw-Hill International: New Delhi, India, 2008.
[2] S. Agrawal, V. Narasayya, and B. Yang, "Integrating vertical and horizontal partitioning into automated physical database design", Proc. of ACM SIGMOD International Conf. Management of Data, Paris, France, pp. 359-370, June 2004.
[3] M.T. Ozsu, and P.Valduriez, Principles of Distributed Database Systems, 3rd edn. Springer Science+Business Media, LLC 2011: New York, USA, 2011.
[4] T. Ulus, and M. Uysal,"Heuristic approach to dynamic data allocation in distributed database systems," Pakistan Journal of Information and Technology, vol. 2(3), pp. 231-239, 2003.
[5] A. Singh, and K.S Kahlon,"Non-replicated dynamic data allocation in distributed database systems," International Journal of Computer Science and Network Security, vol. 9(9), pp. 176-180, 2009.
[6] A. 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, Springer, Singapore, pp. 131-138, 2016.
[7] R. Kumar, and N. Gupta, "Non-redundant dynamic data allocation in distributed database systems," International Journal of Computer Applications, Special Issue on ICNICT, vol.6, pp. 06-10, 2012.
[8] N. Mukharjee,"Non-replicated dynamic fragment allocation in distributed database systems," Springer-Verlag Berlin Heidelberg, CCSIT, Part I, CCIS 131, Bangalore, pp. 560-569, Jan. 2011.
[9] R. Kumar, and N. Gupta, "An extended approach to non-replicated dynamic fragment allocation in distributed database systems," IEEE Xplore, ICICT, Ghaziabad, India, pp. 861-865, 2014.
[10] R. Kumar, and N. Gupta,"Dynamic data allocation in distributed database systems: a systematic survey," International Review on Computers and Softwares, vol. 8(2), pp. 660-667, 2013.
[11] R. Kumar, and N. Gupta, "An extended efficient approach to dynamic fragment allocation in distributed database systems," International Journal of Control Theory and Applications (IJCTA), vol. 9(20), pp. 473-482, 2016.
[12] R. Kumar, and N. Gupta, "Clustered approach to dynamic data allocation in distributed database systems", International Journal of Advanced Research in Computer Science (IJARC), vol. 8(5), pp. 2046-2052, May-June 2017.
[13] A.E.A. Raouf, N.L. Badr, and M.F. Tolba, "Distributed Database System (DSS) Design Over a Cloud Environment", © Springer International Publishing AG, Multimedia Forensics and Security, pp. 97-116, 2017.
[14] A.A. Sewisy, A.A. Amer, and H.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), vol. 13(2), 2017.

AUTHORS PROFILE

Raju Kumar received his master's degree in computer applications from Dr. B.R Ambedkar University, Agra. He is UGC-NET qualified in computer science & applications and pursuing doctorate from department of computer science, faculty of technology, Gurukula Kangri Vishwavidyalaya, Haridwar, India. His research interest includes distributed database focusing on dynamic data allocation, optimization, and soft computing techniques. He published several research papers in international and national journals and conferences. He is a member of IAENG, IACSIT, CSTA, and IEEE society.

Dr. Neena Gupta is assistant professor in the department of Computer Science at Kanya Gurukula Mahavidyalaya, Dehradun (Second campus of Gurukula Kangri Vishwavidyalaya, Haridwar). She did her master's degrees in Computer Applications, and Mathematics. She received her doctorate in Computer Science from Gurukula Kangri Vishwavidyalaya, Haridwar. She is life member of International Association of Engineers (IAENG) and System Society of India. Her area of interest includes distributed database, web database. She published several research papers in international and national journals and conferences.