ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
Vol. 3, Special Issue 20, April 2016

An enhanced methodology for efficient Fragmentation and Re-Allocation in P2PDDBS

C. Balakrishnan
Asst. Professor, Alagappa Institute of Skill Development, Alagappa University, Karaikudi
balasjc@gmail.com

Abstract- The Peer-to-Peer architecture in Distributed Database environment having potential to handle the data in an efficient manner. The design issue is a major one and that has to be addressed with efficient methodology so as to improve the throughput. This work proposed a cluster based Peer-to-Peer architecture named ElasticPeerDB for the distributed databases to address the fragmentation and allocation phases of database design. This work takes the inspiration of the previous works done based on the predicate based fragmentation and introduces the clustering approach for drafting the database architecture and to allocate the fragmented data across the sites.

Keywords: ElasticPeerDB, Peer-to-Peer databases, fragmentation, allocation, priority factor values, clustering approach

I. INTRODUCTION

Distributed database is a collection of multiple, logically interrelated databases distributed over a computer network [1]. This resource distribution improves performance, reliability, availability and modularity that are inherent in distributed systems. There will be a possibility of improved response times to queries and upgrading system capacity or performance incrementally. The sites of a distributed database system may be distributed physically either over a large geographical area (such as the all Indian states), or over a small geographical area such as a single building or a number of adjacent buildings. Distributed database design is one of the major research issues in the area of distributed database system. At the heart of the idea of a distributed system is the distribution of data over multiple sites. The conceptually simplest distribution scheme is to distribute at the table level: any given table is stored in its entirety at some site or it may be partitioned and stored in different sites. A technique of breaking up the database into logical units, which may be assigned for storage at the various sites called data fragmentation. In the data fragmentation, a relation can be partitioned (or fragmented) into several fragments for physical storage purposes and there may be several replicas of each fragment. These fragments contain sufficient information to allow reconstruction of the original relation.

Peer-to-Peer (P2P) technology has no strict definition; it is generally described as having a structure that is contrast to the traditional client-server model. Each node in the network acts as both client and server, requesting data from neighboring nodes as well as routing and serving data for others. The nature of P2P technology makes it well suited for storing multiple copies of data between several nodes, in turn offering reliable access to data and distributing the load of requests. Additionally, the multiple links between nodes make the system more stable as nodes are added and dropped. All the features inherent in P2P technology promise a network that is dynamic, scalable and reliable. Of the several issues in P2P based distributed database environment, the basic and first and foremost problem is to know the location of neighbors [2]. Without the knowledge of the neighbors the unsuccessful queries cannot be transformed across the network to find the appropriate data to execute the query. The above narrated problem is addressed in two ways in general file sharing systems, such as, Chord and Freenet. Since, the Chord and Freenet are widely used in data sharing P2P environments, the characteristics of the two concepts encourage the research directions to include Chord and Freenet in Database environment.

The paper is organized as follows. The next section of this work presents literature reviews of fragmentation, allocation, clustering, Chord and Freenet. Section III describes the ElasticPeerDB architecture. In Section IV implementation details are presented. Finally Section V concludes the paper with future research directions.

II. LITERATURE REVIEW

Most of the research related to fragmentation and allocation has been carried out in the context of relational databases. Navathe et al., [5] proposed vertical partitioning algorithms for database design. Vertical partitioning is done to produce fragments and clustering of attribute columns is done based on the requirements of transactions that closely match. To minimize the transmission cost, a method has been proposed by Peter M. G. Apers [6] to allocate the fragments. The author has presented two different approaches to the allocation management problem and compared the methods for obtaining optimal and heuristics solutions under various ways of computing the cost of an allocation. Lin et al., have done investigation on fragment allocation problem along with an idea of minimizing overall communication cost. An iterative algorithm has been proposed by Lin et al., [7] considering both the physical network and transaction processing strategy.

Eziefe and Barker [8,9,10,11] proposed fragmentation algorithms for class fragmentation both horizontally and vertically in distributed object oriented systems. In their approach to vertical class fragmentation, all frequently accessed attributes and methods of the classes are grouped into a fragment. Navathe [12] has proposed a mixed fragmentation method for distributed database design at the initial level and a mixed fragmentation tool to partition relations using a grid approach. It is based on a graph theoretic algorithm which clusters a set of attributes and predicates into a set of vertical and horizontal fragments, respectively. Karlapalem and Li [13,14] made a study on different types of partitioning schemes in object oriented databases. Horizontal fragmentation algorithm for distributed deductive database systems has been proposed by Lim et. Al [15]. This algorithm handles the horizontal fragmentation by clustering all the tuples in a base relation that are used by queries. Lim and Yiu-Kai Ng [16] presented different approaches for vertical fragmentation of relations and allocation of rules and fragments. It helps to maximize locality of query evaluation and minimizes communication cost and execution time during processing the queries.

Zhou and Sheng [17] tried to solve the vertical fragmentation problem and fragment allocation problem together. Bellatreche et al., [18] proposed two horizontal fragmentation algorithms: Primary algorithm and Derived algorithm. Bellatreche et al., [19,20] made a study on horizontal fragmentation in the object-oriented model. Fragmentation issues related to design of distributed object-oriented databases have been addressed by Malinowski and Chakravarthy [21]. A mixed fragmentation algorithm has been proposed by Baiao and Mattoso [22] for distributed object-oriented databases. Ezeife nad Zheng [23] provided a technique to measure the performance of object horizontal fragments placed at distributed sites and helps to compare horizontal fragmentation schemes. Zheng [24] has defined a technique to initiate dynamic horizontal fragmentation of objects in an object-oriented database system. Huang and Chen [25] proposed a simple and comprehensive model for a fragment allocation problem. Also, they have developed Huang and Chen, two heuristics algorithms to find an optimal allocation of the fragments. Ahmad et al., [26] have addressed the allocation of fragments problem in distributed database system. They have developed a query driven data allocation approach. Various algorithms based on evolutionary computing paradigm have also been proposed by them.

III. ElasticPeerDB ARCHITECTURE

The proposed architecture for ElasticPeerDB follows the clustering of sites based on the locality priority factor factor. The block diagram of proposed architecture is shown in Fig. 1.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
Vol. 3, Special Issue 20, April 2016

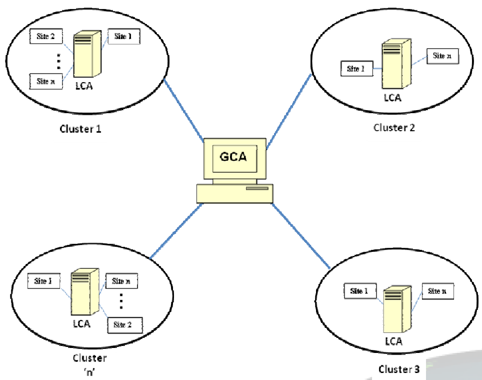

Fig. 1: Clustered approach for ElasticPeerDB architecture

As mentioned in Fig. 1 the sites are clustered with its locality priority factor value and each cluster will be managed by Local Cluster Administrator (LCA) and the whole architecture is administered by Global Cluster Administrator (GCA). The clustering process at the top level (architecture level) is done by the GCA based on the unique number of regions of the sites using Site Information Table (SIT). SIT will give details about each site. It contains information such as site ID, Locality and Region using which clustering of sites is done. The attributes of sites such as, Local Cluster Identification (LCA_id), Site identification (site_id), region of the cluster (cluster_region), Location of the site (site_location) and type of the data stored in that site (site_type) will be handled by LCA of the respective cluster. The global attributes of all clusters like, Global cluster identification (GCA_id), Local Cluster identification (LCA_id) and the region of the cluster (cluster_region) are maintained by GCA of the architecture.

The LCA and GCA are equipped with the functions like, Validator, which validates the relevance of the query. The queries that dissatisfy the criteria expected by Validator will be rejected. Hence, wastage of processing capacity with irrelevant queries is reduced. The LCA Resource Checker finds the appropriate site and data within the cluster and the GCA Resource Checker finds the respective Cluster which owns the required data. The LCA Forwarder, will forward the un-successful queries to GCA and GCA Forwarder will re-direct the query to the appropriate cluster (LCA of the cluster).

The algorithm for fragmentation procedure is given in Fig. 2.

Input:	Total number of sites Site Information Table: LCA and GCA Relation to be fragmented: R Enhanced CRUD matrix: ECRUD
Output:	Fragmented segments
Step 1:	Construct ALPT[R] from ECRUD[R] based on cost functions
Step 2:	Generate predicate set P for the attribute with highest priority factor value
Step 3:	Fragment the relation using the predicate
Step 4:	Cluster fragmented sub-relations.
Step 5:	Re-fragment the fragments based on next highest priority factor valued attribute within the cluster.
Step 6:	Allocate the fragments to sites within the cluster.

Fig. 2: Fragments and Sites Cluster Algorithm

To analyze the performance of the ElasticPeerDB architecture and implementation a distributed banking database system has been taken.

To demonstrate the performance of ElasticPeerDB, initially the number of sites is considered to be ten as shown in Fig. 3.

Fig. 3: Initial number of sites

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
Vol. 3, Special Issue 20, April 2016

Information about each and every site will be given in Site Information Table (SIT) as shown in Table1.

Table 1. Site Information Table (SIT)

Site Id	Locality	Region
1	L1	R1
2	L2	R2
3	L3	R1
4	L4	R3
5	L5	R2
6	L6	R1
7	L7	R3
8	L8	R4
9	L9	R4
10	L10	R1

Based on the initial requisites of ElasticPeerDB architecture, the sites are clustered as follows, there are four unique regions given in the SIT, hence, the ElasticPeerDB is framed with four clusters and the sites are categorized as four groups based on the respective regions of the site. The clustered formation is as shown in Fig. 4 and by taking the derived clusters, the ElasticPeerDB architecture is as shown in Fig. 5.

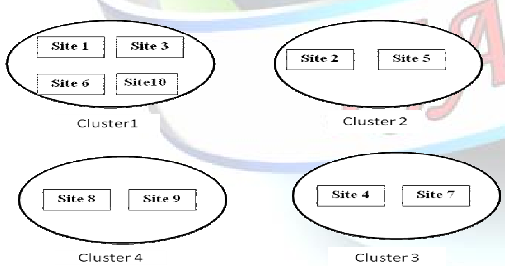

Fig. 4: Clustered framework of sites

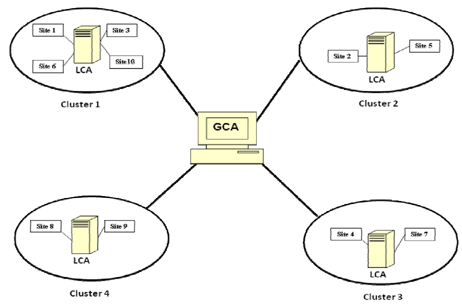

Fig. 5: ElasticPeerDB architecture with four Clusters

The Accounts relation is taken for analyzing the Fragmentation and Allocation in ElasticPeerDB. The attributes and values in Accounts relation is shown in Table 2.

Table 2: Accounts relation

Ano	Category	Cid	Date	Balance	Region
1	A	C1	11/1/15	21000	R1
2	B	C2	21/1/15	13500	R2
3	B	C3	2/2/15	18000	R1
4	C	C4	8/2/15	22000	R3
5	D	C5	24/2/15	3200	R4
6	C	C6	15/3/15	52000	R1
7	E	C7	18/3/15	38000	R2
8	D	C8	28/3/15	11500	R1
9	A	C9	4/4/15	16800	R3
10	A	C10	9/4/15	78000	R1
11	B	C11	11/4/15	23000	R4
12	B	C12	18/4/15	11800	R2

ECRUD matrix should be constructed for the Accounts relation during the requirement analysis phase. From this matrix ALP values will be calculated using the cost functions. For example a sample ALPT for Accounts relation in shown in Table 3.

Table 3: Priority factor values of Accounts relation

Name of Attributes	Priority factor Value
ANO	10
CATEGORY	25
CID	11
DATE	14
BALANCE	18
REGION	58

The highest priority factor valued attribute will be considered as an important attribute for fragmentation. According to that predicate set will

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
Vol. 3, Special Issue 20, April 2016

be generated. For instance, our ALPT shows that Region has the highest priority factor value. So the predicate set will be as follows: P = {Region=R1; Region=R2; Region=R3; Region=R4.

Based on these predicate sets, relation will be fragmented. So we will get the fragments as shown in Table 4.

Table 4: Sub-relation based on predicate 'Regions'

Ano	Category	Cid	Date	Balance	Region
1	A	C1	11/1/15	21000	R1
3	B	C3	2/2/15	18000	R1
6	C	C6	15/3/15	52000	R1
8	D	C8	28/3/15	11500	R1
10	A	C10	9/4/15	78000	R1
2	B	C2	21/1/15	13500	R2
7	E	C7	18/3/15	38000	R2
12	B	C12	18/4/15	11800	R2
4	C	C4	8/2/15	22000	R3
9	A	C9	4/4/15	16800	R3
5	D	C5	24/2/15	3200	R4
11	B	C11	11/4/15	23000	R4

After clustering, re-fragmentation is done on the fragments based on the next highest priority factor value in the ALPT within the cluster. The re-fragmented sub-relations are then allocated to the sites within the cluster as shown in Table 5.

Table 5: Re-fragmented sub-relations allocated to sites in clusters

A1	A2	A3	A4	A5	A6	A7	A8
1	A	C1	11/1/15	21000	R1	1	1
10	A	C10	9/4/15	78000	R1	1	1
3	B	C3	2/2/15	18000	R1	3	1
6	C	C6	15/3/15	52000	R1	6	1
8	D	C8	28/3/15	11500	R1	10	1
2	B	C2	21/1/15	13500	R2	2	2
12	B	C12	18/4/15	11800	R2	2	2
7	E	C7	18/3/15	38000	R2	5	2
4	C	C4	8/2/15	22000	R3	4	3
9	A	C9	4/4/15	16800	R3	7	3
5	D	C5	24/2/15	3200	R4	8	4
11	B	C11	11/4/15	23000	R4	9	4

A1 – ANO A2 – Category A3 – CID
A4 – Date A5 – Balance A6 – Region
A7 – Site A8 - Cluster

If another site is added to any of the clusters, next highest priority factor valued attribute will be taken for further fragmentation.

IV. CONCLUSION

This paper addressed the design requirements for a Peer-to-Peer distributed database. This paper also framed an architecture named ElasticPeerDB. The clustering of sites is done based on the geographical regions of the sites. The clusters are managed by LCA and the overall architecture managed by GCA. Both LCA and GCA are equipped with functions to facilitate the data processing. With this ElasticPeerDB, the data are stored only in sites and the information about sites in a cluster is stored in respective LCA and the information about all clusters is stored in GCA. The relation will be fragmented based on the highest priority factor and those fragments are clustered along with the sites based on common predicate. Within a particular cluster, once again the sub-relation is re-fragmented to allocate data to the sites within the particular cluster based on the next highest priority factor. Finding appropriate data and respective site will be taken care by LCA and GCA. Hence the sites can effectively store and produce results for queries instead of wasting its processing capacity by listening to all queries though the required data are not available in that particular site. The query processing operations are simulated and studied with the results produced with Chord architecture. In future, the query processing and concurrency control mechanisms can be studied in ElasticPeerDB environment.

REFERENCES

[1]. M. T. Ozsu and P. Valduriez, "Principles of Distributed Database Systems", 2nd ed., New Jersey: Prentice-Hall, 1999.

[2]. David R. Karger and Matthias Ruhl, *Finding Nearest Neighbors in Growth-restricted Metrics . ACM Symposium on Theory of Computing (STOC '02)*, Montréal, May 2002

[3]. Chord website: http://pdos.lcs.mit.edu/chord

[4]. Freenet website: http://freenetproject.org

[5]. Shamkant Navathe, Stefano Ceri, Gio Wiederhold and Jinglie Dou, "Vertical Partitioning Algorithms for Database Design", ACM Transactions on Database Systems, Vol. 9, No. 4, pp. 680-710, 1984.

[6]. Peter M. G. Apers, "Data Allocation in Distributed Database Systems", ACM Transactions on Database Systems, Vol. 13, No. 3, pp. 263-304, 1988.

[7]. Xuemin Lin, Maria Orlowska and Yanchun Zhang, "On Data Allocation with the Minimum Overall Communication Costs in Distributed Database Design", ICCI '93 Proceedings of the Fifth International Conference on Computing and Information IEEE Computer Society Washington, 1993.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology
Vol. 3, Special Issue 20, April 2016

[8]. Christie I. Ezeife and Ken Barker, "Horizontal Class Fragmentation in Distributed Object Based Systems", Joint Conference on Engineering Systems Design, pp. 43-52, 1994.

[9]. Christie I. Ezeife and Ken Barker, "Vertical Class Fragmentation in a Distributed Object Based Systems", 1994.

[10]. Christie I. Ezeife and Ken Barker, "A Comprehensive Approach to Horizontal Class Fragmentation in a Distributed Object Based Systems", International Journal of Distributed and Parallel Databases, Vol. 3, No. 3, 1995.

[11]. Christie I. Ezeife and Ken Barker, "Vertical Fragmentation for Advanced Object Models in a Distributed Object Based System", Proceedings of the Eigth International Conference on Computing and Information, 1995.

[12]. Shamkant Navathe, "A Mixed Fragmentation Methodology for Initial Distributed Database Design", 1995.

[13]. Kamalakar Karlapalem and Qing Li, "Partitioning Schemes for Object Oriented Databases", Proceedings of the 5th International Workshop on Research Issues in Data Engineering-Distributed Object Management, 1995.

[14]. Kamalakar Karlapalem and Qing Li, "A Framework for Class Partitioning in Object-Oriented Databases", Distributed and Parallel Databases, Vol. 8, No. 3, 2000.

[15]. Seung-Jin Lim and Yiu-Kai Ng, "A Formal Approach for Horizontal Fragmentation in Distributed Deductive Database Design", Proceedings of the 7th International Conference on Database and Expert Systems Applications, 1996.

[16]. Seung-Jin Lim and Yiu-Kai Ng, "Vertical Fragmentation and Allocation in Distributed Deductive Database Systems", Information Systems, Vol. 22, No. 1, pp. 1-24, 1997.

[17]. Zehai Zhou and Olivia R.Liu Sheng, "Vertical Data Fragmentation and Fragment Allocation in Distributed Database Systems", Proceedings of PACIS, 1997.

[18]. Ladjel Bellatreche, Kamalakar Karlapalem and Ana Simonet, "Horizontal Class Partitioning in Object-Oriented Databases", Proceedings of the 8th International Conference on Database and Expert Systems Applications, 1997.

[19]. Ladjel Bellatreche, Kamalakar Karlapalem and Ana Simonet, "Horizontal Class Partitioning for Queries in Object-Oriented Databases", Technical Report HKUST-CS98-6, 1998.

[20]. Ladjel Bellatreche, Kamalakar Karlapalem and Ana Simonet, "Algorithms and Support for Horizontal Class Partitioning in Object-Oriented Databases", Distributed and Parallel Databases, Vol. 8, No. 2, pp. 155-179, 2000.

[21]. Elzbieta Malinowski and Sharma Charavarthy, "Fragmentation Techniques for Distributing Object-Oriented Databases", Proceedings of the 16th International Conference on Conceptual Modeling, 1997.

[22]. Fernanda Baiao and Marta Mattoso, "A Mixed Fragmentation Algorithm for Distributed Object Oriented Databases", Proceedings of the Ninth International Conference on Computing Information, 1998.

[23]. C.I.Ezeife and Jian Zheng, "Measuring the Performance of Database Object Horizontal Fragmentation Schemes", In Proceedings of the 3rd IEEE international database engineering and Applications Symposium, 1999.

[24]. Jian Zheng, "Towards Making Object Horizontal Fragmentation Dynamic", The University of Windsor, MS thesis, 1999.

[25]. Yin-Fu Huang and Jyh-Her Chen, "Fragment Allocation in Distributed Database Design", Journal Of Information Science And Engineering 17, pp. 491-506, 2001.

[26]. Ishfaq Ahmad, Yu-Kwong Kwok and Siu-Kai So, "Evolutionary Algorithms for Allocating Data in Distributed Database Systems", Distributed and Parallel Databases, Vol. 11, 2002.

BIOGRAPHY

Dr. C. Balakrishnan is working as Assistant Professor in Alagappa Institute of Skill Development, Alagappa University for B.Voc. Software Development programme. Previously worked in St. Joseph's College (Autonomous), Tiruchirappalli. Published 25 research articles and 4 books in National / International arena and cleared both NET and SET examinations.