THINK INDIA JOURNAL

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

Distributed Database Systems Design Challenges and Countermeasures - A Study

Sashi Tarun
Assistant Professor, School of CSE
Lovely Professional University
Phagwara, Punjab, India

Dr. Ranbir Singh Batth
Associate Professor, School of CSE
Lovely Professional University
Phagwara, Punjab, India

Abstract

Database systems and computer network make easy distribution of data into the geographical dispersed locations. Distribution of data at network nodes enables us to sustain data availability, reliability and achieve performance. A continuous effort were given by researcher with the passage of time to cope-up with existing design weaknesses and seems to be a challenging task due to expandable behavior of the network, continuous redesign of existing distributed system to entertain queries of different nature from different users. To implement distributed system a sequential process of fragmentation, allocation and replication of data is required. In a distributed transactional environment getting relevant data and satisfying query responses are the key issues. So, appropriate ramification in the existing approaches is required to overcome distributed data distribution challenges. In my propose paper, it is to focus on highlighting different challenges of fragmentation, allocation and replication in distributed environment with their countermeasures to resolve it.

Keywords: data distribution, fragmentation, allocation, replication, availability, reliability, redesign

---

INTRODUCTION

In distributed system, it is quite overhead to avail intended data from more than one sites as it leads to low query performance and high communication cost. Optimal partitioning, allocation of data and uniformity of data at each network site by replicating all data changes dynamically are the sole purpose and is helpful to achieve data consistency, availability and reliability of data. All above parameters can be achieved by follow an optimal solution to fragmentation, allocation and replication of data. User's query patterns and their results are depends on the fragmented data distributed over different remote sites. A good distributed database fulfill required data needs of users through their queries apply during his mobility on mobile host or his fixed host. In this paper, focus is to highlight different challenges of distributed database systems as data fragmentation, allocation and replication process. Fragments are those logical units responsible

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

to satisfy user data requirements based on queries supplied. User queries can be classified into four general categories on the basis of query-of- interest:

a. **Queries on common data-items:** Mostly queries used for the extraction of data are from group transactions (schedules) and are based on access of common data-items.
b. **Multi-database queries:** Some queries involve extraction of data records collectively and selectively from multi-databases.
c. **Queries for partially data access:** Queries to access the data records partially using database views. Some users have limited access permission and have access on partial data. So, these users limited up to views rather than the original table.
d. **Queries based on granularity:** Queries to access data records based on granularity of data-items. Based on granularity user's has two choices to get the required data or information:
    a. To access the selected data-items (fine-grained) from the global schemas.
    b. To access the whole data-block (coarse-grained) from the global schemas to satisfy queries.

Based on above stated data queries categories, distributed database design should be able to handle outside workforce by fast query response, easy data availability, provide data reliability features in case of fault, failure or error.

1.1. Environmental Classifications of Queries

Mobile environment allow users to deals with different database queries and initiate, execute and complete their queries in different modes:

a. **Queries on Fixed Host (FH**): Here mobile host is act as a thin client. Here users generally initiated his query from the mobile host but executed and completed at fixed host.
b. **Queries on Mobile Host (MH):** Here database queries are initiated, executed and completed by the mobile host itself. It is happen only when requested data is available physically at requested site. To achieve such functionalities mobile host should have high processing and storage capabilities and responsible to manage the data fully.
c. **Queries on MH and FH:** Here database queries are entertained by joint effort of both the host (MH and FH). It is helpful to execute different sub-queries on both hosts.
d. **Queries among several MHs:** Here database queries are entertained by multiple mobile hosts. It is all about implementing of strategy called peer-to-peer computing. Here all mobile hosts are equal and none of them are superior to each other and coordinate to each other to execute the distributed query.

1.2. Fragmentation of Data

It is a process through which large global schema is partition into individual units at geographical locations. Satisfaction level of query is depending on fragment units contain different data-items. Partition should be designed as according to changing behavior of the database users. Every fragment designing should be in such as way that database can be

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

reconstruct by combining different fragments together. Fragments are designed to satisfy different aims:

a. **Reliability:** Designing of fragments is one of the challenging task and to decide about the inclusion of different data-items to build a fragment. If fragments is able to satisfy user queries than reliability can be achieved.
b. **Performance:** Query on large global schema for required data affects the performance. As query includes data-items from multi-database, multiple relations and sometimes complex queries reduce the query response. To handle such issues fragmentation play an important role as necessary data records can be easily available from fragments without delay.
c. **Balanced Storage Capacity:** Fragments are small logical units of a large database. Fixed size block storage is assigned for each fragment. Each fragment is having data-items of user interest can be achieved using fine grained approach. It eliminates unnecessary inclusion of items so that storage/size of the fragment can be easily maintained.
d. **Communication cost:** Availability of data from short path reduces the communication cost. Availability of data by user query depends upon shortest path and can be achieved by proper placement of data-items into fragments.

1.2.1. Data Fragmentation Challenges

In present scenario, data is increasing at a rapid rate and are being populated into the database by different sides or corners. Some of the data are generally accumulated from different sources i.e. web applications interfaces, databases in the form of excel sheets, and populated from daily business transactions gathered from the outside world. Data crowdedness affects complexity in database, consistency of data, maintenance and its security issues in the database. As a result of this every organization looking into centralized large data volumes and think about how to handle and make useful this vast data or information for mobile users. Different questions are comes into mind for a solution to handle such crowded data:

a. How to split the large database into fragments?
b. How to allocate fragments into different location, so that to locate them easily.
c. And to decide about fragments to replicate.

Fragmentation is a process by which large schema records are divided into independent pieces or parts to increase the speed of query processing. Fragmentation process further results to allocation and replication of data in geographical dispersed sites. Fragmentation enables us to achieve complexity issues, maintenance problems, and relevancy of data issues in the fragments. Different challenges are faced by the database designer during database fragmentation:

a. **Weak policies for data:** Very few organizations have policies regarding the placement of data at node level. Usually due to lack of decision about the selection of data which can or cannot be included to design such systems. Nor they know how fragmentation is applied on the data with optimal results.

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

b. **Suitability of required data-items:** Problems occur when users wish to extract data-items by coarse-grained granularity approach (where whole data blocks of requested data-items are loaded into the database buffer cache) to satisfy different database queries. In such cases, fragmentation seems difficult to achieve due to un-suitability of data-items. Inclusion of miscellaneous data-items in the database queries is not helpful to achieve user satisfy level.
c. **Hindrance in distributed data maintenance:** Maintenance problems exist when queries extract data from more than one database. In an organization, where inter-related data-items from multi-database are used to satisfy user queries affect maintenance of distributed data.
d. **Inclusion of data-items:** Existing strategies of fragmentation are not able to wrap each data-item to satisfy user queries as a whole. It is because fragment contains selected records from few relations and is small in size. But when a complex query is applied against fragments, its speed of execution (performance) becomes slow due to non-availability of data-items at node where transactions occur.
e. **Incompleteness:** Before applying fragmentation on global schemas, check whether schema is complete or final for distributed fragmentation or not. In case if the schema is incomplete then chances of re-fragmentation occur.
f. **Using of multi-database joins in query:** Fragmentation becomes failure, if data items are from more than one relation in a multi-database environment. In such case fragments data is of no use and stress is given to control the use of distributed joins. Most of the user's queries access data from single or multiple data relations belongs to single database.
g. **Correctness issue:** Integrity checking is not possible in fragments because integrity can be checked inside the relation from where the attributes belongs. Fragments contain set of attributes of single database or from multiple databases, so integrity checking is not possible and correctness of data is not easy.

1.3. Allocation of Data

With a view to allocate data (i.e. fragments) into different network nodes, it is necessary to place the data patterns at different nodes in such a way that users easily get the required data or information without looking into multiple network sites. Availability of data can be from the requested node or from its adjoining connected nodes without the involvement of far nodes. Fragments allocation into distributed system in such a way that it should satisfy the following conditions:

a. Transactions performed by the mobile users (in distributed system) and normal database users (in DBMSs) are independent to each other. It is all about execution of global queries. They do not affect the functioning of query processing and data extraction from both the sides.
b. The size of data-items to be fragmented and number of replicas should be in position to accommodate or entertain each new data-item to make distributed system flexible to adapt changes.
c. Allocation of data into geographical sites allows executing different modes of distributed queries in a mobile environment.

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

1.3.1. Data Allocation Challenges

a. **Size:** Number of fragments on individual sites is also affecting the data allocation. Engagement of multiple fragments required more memory space, and results into multiple involvements of users which raise heavy load due to high query execution frequency on fragments and affect the performance.
b. **Ambiguity in fragments:** During fragment allocation it is to ensure that no more redundancy exists in sites causes ambiguity in queries result.
c. **Fragment scatterings:** Availability of data-records from different sites causes performance degradation. So, allocation of the fragments should be in such a way that data records can be easily obtained from single or adjoining sites from the activity center.
d. **Dependency:** Dependency among fragments also affects allocation strategies, so relations among fragments no more exists as it increases the complexity of a data allocation.
e. **Weak data placement policies:** Unplanned placement of data-items in fragments affects the data allocation strategy. Distributed transactions among several MHs are not fulfilled because data required to complete the query result is not available at nearest location. On the other hand, demanded data is not available completely at single site.

1.4. Replication of Data

With a view to maintain high data availability and consistent data at every node, it is necessary to replicate the data changes occur in the network nodes. It can be achieve by the involvement of wireless network technology, which made feasible to access the required data from any of the corner within the network or expansion. Here concern is to provide consistent data to every network users to avoid differences in data at each node. To do this, there is a need to work on dynamic replication strategy responsible to maintain identical copies of data everywhere.

Every organization depends on their existing data, somewhere stored in the network to satisfy their business needs. Data being an asset or a liability, and it's availability to every users in the network is always on high demand. It is flexible to work with data in centralized system but once it is deputed into diverse distributed network nodes challenges comes in different ways about the correctness of data. Some viewpoints in this regards are following as:

a. Availability of data with high consistency rate can be achieved by the replication strategies or not.
b. Data requirement of the user's is dynamic in nature. Is new replication strategies are in position to adapt with the current pace of changing user database queries?
c. How data can be identical, in case of failure (i.e. network failure) during the data replication/propagation process.

1.4.1. Data Replication Challenges

In replication different challenges are in the form of:

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

a. Partially replication of data, due to not decided about the amount of data is being replicated to geographical sites in a concurrent environment.
b. Non-availability of participating nodes due to disconnection or failure causes data propagation errors problems.
c. When changes on common data-items took place simultaneously from mobile users in a concurrent environment rise lost-update problem issue. It is a situation in which only one user changes are reflected on the common data-item. At this situation replication process is incomplete or useless because it allows propagating changes of one of the user in the database.
d. In coarse-grain granularity, when changes take places on few data-items at any activity center, whole data blocks changes are propagated to other sites. As a result it seems delay in data propagation. To reduce propagation delay fined-grained granularity of data-item is used in which only changed data-item are propagated to avoid delay.
e. Replication process is said to be in-complete or partial if data records are not found identical in comparison to other sites because of low signal strength.
f. Another problem occurs in the form of propagation delay. It refers, delay in the progression of propagation as the size of the network grows.
g. Lack of information about the dispersed network sites to make replication process affective.

LITERATURE SURVEY

Work on distributed Resource Description Framework (RDF) is performed to manage the growing massive RDF. To utilize this large volume RDF is partition into small parts called fragments and further approach the same for allocation in the distributed database environment. Here, focus is given to reduce the communication cost during the query processing tasks. It also ensures to maintain data integrity and approximation ratio due to frequent access patterns from outside. Here RDF graph is divided using three fragmentation strategies namely horizontal, vertical and mixed fragmentation based on frequent access patterns. It is also focus on balancing and allocation of fragments into different sites [12].

A systematic review on data distribution strategies i.e. Data fragmentation, Allocation and Replication is performed by the author. In this paper, it is indicated that different problems are faced by the designers during using and designing of these strategies. First, Data fragmentation is having problems of Join Optimization, when query trying to combine more than one fragments from more than one geographical site to fetch the required data. It reduces the response time. Secondly, includes data allocation problem about finding optimal technique helpful to allocate fragments to different sites.

A heuristic approach for fragmentation is proposed to reduce transmission cost (TC) of queries in distributed environment. Here, at initial stage fragmentation is based on cost-effective model in context of relational model and at later stage based on DDBS design. Different replication based

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

allocation scenario were proposed i.e. mixed replication-based data allocation scenario (MAS), full replication-based data allocation scenario (FAS), and non replication data allocation scenario (NAS)[7].

A modified Bond Energy Algorithm (BEA) is proposed and it is a hierarchical process to make fragments vertically and allocate the fragments into geographical sites across the network. This algorithm use affinity of attributes and is helpful to generate cluster of attributes, to calculate cluster allocation cost and also decide about their appropriate sites for allocation. Here attributes accessed collectively by the same query are placed into one fragment [5].

A study was to review and compare the existing algorithms in design perspective with a view to identify their strength and weakness. This is just to present an affective design for the distribution of data fragments on the distributed environment 6]. A non-redundant dynamic fragment allocation technique is proposed and is based on the changing access pattern at different sites with a view to improve the performance. Here fragments reallocation is depend on access made on each fragment data volumes based on defined time constraint and threshold value. This proposed technique change the reallocation strategy by modifying the read and write data volume factor and introduced threshold time volume and Distance Constraints Algorithm. Write data volume is considered for the reallocation process when more than one sites approach for the fragments. This ensures the overall improvement of distributed system performance [11].

A hybrid optimized model using information on the type and frequency of queries for fragmentation of data horizontally and vertically and is based on supervised machine learning approach to produce non overlapping fragments. These fragments are maintained by archiving process rather than deletion operation on them. These fragments are used to facilitate searching operations based on index so that database tables are partition horizontally and vertically [8].

Two algorithms Modify Create Read Update Delete (MCRUD) and Matrix based Fragmentation (MMF) for efficient partitioning of large databases without query statistics. It shows that earlier approaches of partitioning were based on type and frequency of the queries called observed or experimental data. Here it is also indicated that earlier partitioning approach were not suitable because at the initial stage of the design of distributed database query statistics are not available. In his paper, an optimal fragmentation technique is proposed to partition global relations of a distributed database when there is no data access statistics and no query execution frequencies are available. When data access statistics and query execution frequencies are not available at the initial stage then MMF is responsible to partition relation in the distributed database. MCRUD is responsible to take fragmentation decision without using empirical data [14].

Work on different replication strategies in MANET, mobile database, distributed database, and cellular network etc is highlighted. It discuss about replication protocols as ROWA, ROWA-A and Quorum Based Protocol. All are replica control protocol, ROWA is responsible to fetch the read request values to the nearest site from the occurrence of request location and replicate the

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

changes to all the sites. An alternative approach is in the form of ROWA-Available and is same as ROWA in the case of read operations but replicate the changes only to all available replica copies and do not bother about any replication failure. ROWA-A is responsible for maintaining the availability of data but do not compromised with the correctness of data. In case of failure users are working with stale value of data. It shows incorrect or out-of-date copy of replica. Quorum based replica is to update the subset of replicas rather than replicate the changes as a whole and helpful to maintain consistency of data [2].

An integrated approach is proposed for DDBMS namely data fragmentation, network sites clustering and allocation of fragments. This work is responsible to improvise problems in the form of; fragmentation, redundancy in data allocation and redistribution problem due to complexity, to maintain data availability and consistency issues [14].

It is also highlighted, to maintain inconsistency issues faced by the mobile users during the access of database in his mobility from any of the activity center. Here a 5-cube structure with nearest-neighbors propagation distribution protocol is proposed to make useful distributed database system for the mobile users. It ensures consistent data to all mobile users/sites by dynamically replicates the changes to all its adjoining sites from the transactional node [16].

A new dynamic de-allocation approach for a given fragment as Update Matrix (UM) and Distance Cost Matrix (DM) is proposed. It works on the basis of changing data access patterns in replicated and non-replicated distributed database system. It was assumed that fragments are allocated on network site is based on applied frequency value of the database data items. Reallocation of data fragments on the remote sites is planned based on communication and update cost value. Each fragment is having update cost value. Fragment having maximum update cost value is considered for reallocation and chosen candidate site to store fragments to minimize the communication cost. UM is defined as the value getting after issuing of update query at a particular site for the manipulated fragment. In this approach when same query is applied at more than one site, then queries can be treated different to each other and having different frequency value [4].

An algorithm called Simulates Annealing with Genetic Algorithm (SAGA) is used for optimal allocation of fragment in distributed environment. Here, allocation of data is depends on access patterns for fragments and focused on reducing the allocation cost during movement of data fragment from one site to another [3].

A problem in large scale mobile distributed database system is introduced by replication architecture for distributing replica and updates propagation protocol to propagate recent changes occurred in different objects in distributed system to achieve the data consistency. In his paper presented a new binary hybrid approach consists of pessimistic and optimistic replication strategy, helpful to work with large number of replicas and reduce the data inconsistency rate to support mobile users. This replication is helpful to achieve update propagation delay reduction

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

and less communication cost. Here replication architecture is proposed consist of master, zone and cell level and Wheel based updates propagation protocol is used to maintain replicas in mobile distributed database [1].

A decentralized approach for dynamic table fragmentation and allocation in distributed database systems is proposed. It is based on observation and monitoring of the sites access patterns to tables which reforms fragmentation, replication, and reallocation based on recent access history aiming at maximizing the number of local accesses compared to accesses from remote sites [9].

A new technique called Attribute Level Precedence (ALP) to partition global schema/database relations at initial and later stage in case of non-availability of data access statistics and query execution frequencies. ALP technique is capable to take advance decision for fragmentation at the initial stage (i.e. knowledge gathered during requirement analysis phase) without empirical data statistics. ALP is a table responsible to fragment a relation horizontally based on the importance of an attribute in a network site [15].

DATA DISTRIBUTION COUNTERMEASURES

Following measures are to be taken to handle fragmentation, allocation and replication challenges include:

Measure 1 (for fragmentation): FP-Growth method of association rule mining can be used to discover frequently used data-items/attribute combinations from the database user queries. It helps to generate strong association rules from the frequent item sets; these rules must satisfy minimum support and minimum confidence. This approach is helpful to design data fragments in an optimal way to achieve required data requirements of the users. FP-growth responsible to mine frequent patterns without follows the candidate key generation. Following steps are required to satisfy measure 1:

a. Generation of transaction details based on data access pattern on the relations. Every transaction is having the details of used data-items at each query level.
b. Based on the support count threshold, check out the occurrence or frequency of each data-item.
c. If calculated frequency < threshold (support count) value.
d. Continue with data-items equal or greater than threshold value.
e. Write the frequent item sets in descending order of their support counts.
f. Construct Frequent Pattern Growth tree for the generation of frequent patterns.
g. Check for the least support count
h. Then divide the compressed database into a set of conditional databases, each associated with one frequent item or “pattern fragment”, and mines each such database separately.
i. At the end frequent patterns of different item sets combinations are discovered.

Measure 2 (for achieving performance after fragments allocation and in replication process): Data records can be extracting from single site completely or from adjoining sites in

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

case of partial availability of data at fragment level using Intelligent Agents (IA). There is no need to move every node or site in a hierarchy to satisfy user query requests. Intelligent agents are proficient of interact with other agents through compromise and /or collaboration, to assure propose objectives.

Different properties of intelligent agents make them functional includes:

a) Intelligent agent freely moving around any of the network nodes and communicate to each other to satisfy any data requirement.
b) IA trustworthy behavior as agent does not provide or communicate false information to the network nodes in case of data requirements.
c) IA always works for achieving the decided task for which they are meant for without any conflicting errors. Agent will therefore always try to do what is asked of it.
d) Agent believes in achieving of goals based on his beliefs or wisdom permit.
e) Agents are having learning and adaptive behavior and improve performance over time

Data replication is an approach to build a system where requirement of high scalability and data accessibility is required. To achieve the replication measures, data changes at each node need to be simultaneously synchronized with the help of mobile agents. The use of mobile agent technology is able to achieve distributed data management in the network of nodes spread over dispersed system with the available system resources.

CONCLUSION

It is conclude in this paper that data distribution task is core task of the distributed database system and begins by dividing the table into independent segments, its allocation into different geographical sites and work for the effective data replication results. In spite of given efforts from researchers for optimal results some challenges are still there and looking for better fragmentation, allocation strategy and replication scheme. In this paper different challenges are highlighted during the fragmentation, allocation and replication and given countermeasure for the same for better system.

Data distribution challenges are handling by applying frequent pattern-growth method of association rules mining and by the use of intelligent agents at node level to achieve performance and get to know about the data copies at individual sites to perform replication process.

REFERENCES

[1] A. Ahmad, “A Novel Replication Strategy for Large Scale Mobile Distributed Database Systems”, Journal of Engineering Science and Technology, Vol. 6, No. 3, pp. 268-299, 2011.
[2] D. Rane, M.P Dhore, “Overview of Data Replication Strategies in Various Mobile Environment”, IOSR Journal of Computer Engineering, pp. 01-06, 2016.

ISSN: 0971-1260
Vol-22-Issue-17-September-2019

[3] H. I. Abdalla, "An Efficient Approach for Data Placement in Distributed Systems", Fifth FTRA International Conference on Multimedia and ubiquitous Engineering, pp. 297-301, 2011
[4] H. I. Abdalla, "A New Data Re-Allocation Model for Distributed Database System", International Journal of Database Theory and Application, Vol. 5, No. 2, pp. 45-60, 2012.
[5] H. Rahimi, F. Parand, "Hierarchical Simultaneous Vertical Fragmentation and Allocation using Modified Bond Energy Algorithm in distributed databases", Applied Computing and Informatics (14), pp.127-133, 2018
[6] H. A. Fuaad, A. A. Ibrahim, A. Majed and A. Asem, "A Survey on Distributed Database Fragmentation, Allocation and Replication Algorithms", British Journal of Applied Science and Technology, 27(2): pp.1-12, 2018
[7] H. Abdalla, A. M. Artoli, "Towards an Efficient Data Fragmentation, Allocation, and Clustering Approach in a Distributed Environment",Information, Vol 10, No.112, 2019
[8] I. B. Oriji, I.C. Ejiofor, "A Hybrid Model for Data Fragmentation in Distributed System", International Research Journal of Computer Science, Issue 04, Volume 5, pp. 186-192, 2018
[9] J. O. Hauglid, N. H. Ryeng, K. Norvag "DYFRAM: Dynamic Fragmentation and Replica Management in Distributed Database Systems", Distributed Parallel Databases, pp 157–185, 2010.
[10] K. K. Ezechiel, S. K. R. Agarwal, "A Systematic Review on Distributed Databases Systems and their Techniques", Journal of Theoretical and Applied Information Technology, Vol. 96, No. 1,236-266, 2019
[11] N. K. Z. Lwin, T. M. Naing, "Non-Redundant Dynamic Fragment Allocation with Horizontal Partition in Distributed Database System", ICIIBMS, Bangkok, Thailand, pp.300-305, 2018
[12] Peng Peng, Lei Zou,Lei Chen,Dongyan Zhao,"Adaptive Distributed RDF Graph Fragmentation and Allocation based on Query Workload", IEEE Transactions on Knowledge and Data Engineering, Vol. 31, No. 4, pp. 670-685, April 2019
[13] Rizik M. H. Al-Sayyed, D. Suleiman, M. A. A. Itriq, I. Hababeh, "A New Approach for Fragmentation and Allocation to Improve the Distributed Database Management System Performance", Journal of Software Engineering and Applications, Vol. 7, pp. 891-905, 2014.
[14] S. Tarun, "A Reputation Replica Propagation Strategy for Mobile Users in Mobile Distributed Database System", International Journal of Grid and Distributed Computing, Vol. 5, No. 4, 2012.
[15] S. I. Khan, "Efficient Partitioning of Large Databases without Query Statistics", Database System Journal, pp. 34-53, 2016.
[16] S. I. Khan and A. S. L. Hoque, "A New Technique for Database Fragmentation in Distributed System", International Journal of Computer Application, Vol. 5, No. 9, pp. 20-24, 2010