Database Replication in DDBMS

Tunir Roy
Department Of Computer Application
Kalyani Government Engineering College
tunir27@gmail.com

Suhrid Ranjan Das
Department Of Computer Application
Kalyani Government Engineering College
suhriddas1@gmail.com

Abhro Bhattacharjee
Department Of Computer Application
Kalyani Government Engineering College
abhrobhattacharjee@gmail.com

Pradipto Karmakar
Department Of Computer Application
Kalyani Government Engineering College
pradiptokarmakar95@gmail.com

***Abstract*—Data Replication has been a major concern for data warehousing and data storage. Many different data replication procedures have been devised such as JPaxos etc. We in this paper formulate a new replication algorithm which takes into account the network in which the distributed sites are present. In doing so we try to establish whether our algorithm is more energy efficient and whether it gives better power efficiency than just using traditional data replication techniques.**

I. Introduction

Data Replication is the process of storing data in one or more site or nodes. It is useful in improving the availability of data. It is simply copying data from a database from one server to another server so that all the users can share the same data without any inconsistency. The result is a distributed database in which users can access data relevant to their tasks without interfering with the work of others. Data replication encompasses duplication of transactions on an ongoing basis, so that the replicate is in a consistently updated state and synchronized with the source. However in data replication data is available at different locations, but a particular relation has to reside at only one location. There can be full replication, in which the whole database is stored at every site. There can also be partial replication, in which some frequently used fragment of the database are replicated and others are not replicated.

II. Types of Data Replication

A. Transactional Replication

In Transactional replication users receive full initial copies of the database and then receive updates as data changes. Data is copied in real time from the publisher to the receiving database (subscriber) in the same order as they occur with the publisher therefore in this type of replication, *transactional consistency is guaranteed***.** Transactional replication is typically used in server-to-server environments. It does not simply copy the data changes, but rather consistently and accurately replicates each change.

B. Snapshot Replication

Snapshot replication distributes data exactly as it appears at a specific moment in time does not monitor for updates to the data. The entire snapshot is generated and sent to Users. *Snapshot replication is generally used when data changes are infrequent*. It is bit slower than transactional because on each attempt it moves multiple records from one end to the other end. Snapshot replication is a good way to perform initial synchronization between the publisher and the subscriber.

C. Merge Replication

Data from two or more databases is combined into a single database. Merge replication is the most complex type of replication because it allows both publisher and subscriber to independently make changes to the database. Merge replication is typically used in server-to-client environments. It allows changes to be sent from one publisher to multiple subscribers.

III. MOTIVATION

The above mentioned replication techniques work with two phase commit protocol and it's likes where the sites communicate with each other with a fixed set of rules to maintain the *ACID* property of the transactions. In this paper we work with the energy respect in the communication of the sites and how communication between the sites can be made more energy efficient. To make the whole process more streamlined and energy efficient we take in the concepts of the LEACH Protocol of the wireless sensor networks (WSN) and try to modify it. The working of our abovementioned modified algorithm is mentioned below.

IV. WORKING PRINCIPLE

A. Defining Nodes:

The sites of the DDBMS hereby referred to as nodes typically consists of:

a) Position: The positioning of the node in 2D coordinate system .

b) Tables: The detailed schema design of the node.

c) Communication range: The maximum range of the node to be able to communicate with other nodes.

d) Energy: The energy level of the node at a certain point to be able to communicate.

Example of a node: Position = [130, 120]
Tables = ['tid', 'address']
Communication Range = 30
Energy = 100

B. Rule base of the algorithm

After the node space has been defined the algorithm clusters the node space based on the following criteria.

- The node with *highest communication range* is chosen as the initial cluster head.

- The initial cluster head selects members based on the follow two criteria:-
    - Distance of node from cluster-head<=communication range of cluster-head.
    - Schema similarity factor>=1
- After the node space has been clustered the initial cluster-head now calculates whether it is the best cluster-head for its cluster. It does this by:-
    - Checking if any other node has lesser distance measure from all other nodes in the cluster.
    - Checking if any other node has greater schema similarity factor than the initial cluster head.
- After the cluster head has been finalized then the cluster head serves each request that occurs and no other nodes communicate with each other.

C. Clusterhead Selection and Cluster Formation:

Firstly the node/site space is defined by the DBA. An initial cluster head is then chosen among the various nodes present depending solely upon the highest communication range of the nodes. Once the initial cluster head has been chosen we move on to defining the cluster of the cluster head. The member nodes of the cluster is then formed based upon two parameters: a) Euclidean Distance of the node from the cluster head and b) Intersection of the schema of the nodes with the cluster head. Keeping the initial cluster head fixed, all the other nodes are visited(communicated to send their details to the initial cluster head). If the nodes fall within the communication range of the cluster head, then they are chosen to be the members of the cluster. This cluster member list is then further filtered using schema matching factor. This is done so that dissimilar schema nodes are not present in a cluster, increasing the communication overhead. After the cluster members of the cluster has been obtained we proceed to check whether the selected cluster head of that cluster is optimal.

To check whether the selected cluster head is optimal, we calculate whether some other node in the cluster has *lesser* intra clustural distance between the selected node and other nodes present in the cluster and whether the selected node in question has *greater* schema intersection count than the initial cluster head respectively. If the above two condition satisfies then the initial cluster head sends a signal to the node which is deemed to be the optimised cluster head promoting it to be the new cluster head and demoting itself to a member node of thr cluster. Thus the optimal node is given by the initial cluster head itself reducing additional communication overhead to compute the optimal cluster head and saving energy in the process. In the above way the node space is clustered in it's entirety such that each node belongs to a cluster with its respective cluster head.

D. Communication between the sites

Communication between the sites happens when there is a need to update the content of any site and the respective schema is present in other sites as well. As it is a DDBMS data update request may come to any site and then it needs to be done in the sites containing the schema. Communication may also occur when the data replication needs to be done to improve the availability of data. The energy lost in each such type of communication is defined by:

$Energy = (\text{Distance})^2 \text{ X const}$

Where const = 0 .0009 is arbitrarily taken.

Communication can only occur between two cluster heads and not between individual nodes. Any communication between the nodes must be done with the help of the cluster head nodes to which they belong. Let communication occur between two nodes say A and B and the request initiated at A. Now two cases can occur:

a) Both the nodes belong to the same cluster.

b) The nodes belong to different clusters.

In the 1st case the node A must initiate contact with the it's cluster head node say AC and AC then communicates with B so that the changes made in A can be reflected in B also. The energy spent in this is obtained by the *Energy* equation. In second case let the cluster heads of A and B clusters be AC and BC respectively. A initiates communication with AC and AC then communicates the information with BC. BC then sends the information with B where B updates its schema accordingly. Thus in all single cluster hop communication takes place where between A and B' cluster head AC and BC must communicate with each other directly and not use another cluster head to relay the information.

E. Reclustering of the cluster members:

All the above processes requires energy and it may so happen that nodes may run out of energy and need to be recharged to continue functioning. The nodes remains inactive during this recharging process and after desired state of energy has been reached it becomes active again. When nodes become inactive the cluster space of the scenario changes and thus new clusters have to be formed. This is also required when individual cluster nodes goes inactive and a new cluster head needs to be promoted among the cluster members. Thus in any moment of time the node space in its entirety is always clustered and each cluster has a corresponding cluster head.

V. RESULTS

Initially the node space is been defined as follows where each nodes represents a site in DDBMS architecture. The nodes are initially initialized with 100 energy and all of them have their cluster head status set as 0. In this way 12 nodes are defined which are shown in Table 1.

Sr No	NODE SPACE		
	Position	***Tables***	***Communication Range***
1.	[50, 50]	['id', 'roll', 'name']	15
2.	[20, 30]	['id', 'roll']	30
3.	[100, 0]	['id', 'address']	30
4.	[30, 70]	['id', 'name']	30
5.	[40, 50]	['id','name','roll','marks','section']	40
6.	[70, 70]	['id', 'marks', 'section']	30
7.	[120, 20]	['id', 'lang']	100
8.	[100, 100]	['tid', 'school']	30
9.	[110, 130]	['tid', 'school', 'teacher', 'salary']	150
10.	[120, 110]	['tid', 'school', 'teacher', 'salary', 'address']	55
11.	[130, 120]	['tid', 'address']	30
12.	[90, 120]	['tid', 'subject']	30
13.	[30, 20]	['sid', 'staff_name']	150

After the node space has been defined the clustering starts and the node with the first largest communication range, *Node 0*, is set to be the initial cluster head. *Node 0* then broadcasts a signal to all the nodes present in its communication range. The nodes in turn send an acknowledgement signal along with their position and schema details to the initial cluster head *Node 0*. Now the cluster head calculates the *distances* and the *schema similarity/intersection* factor and select the best suited nodes to form its cluster. After the initial cluster and its members

Cluster No	**NODE SPACE AFTER CLUSTERING**				
	Nodes	***Position***	***Tables***	***Communication Range***	***Energy***
Cluster 1	**0**	**[50, 50]**	**['id', 'roll', 'name']**	**150**	**49.24**
	1	[20, 30]	['id', 'roll']	30	98.83
	2	[100, 0]	['id', 'address']	30	95.5
	3	[30, 70]	['id', 'name']	30	99.28
	4	[40, 50]	['id', 'name', 'roll', 'marks', 'section']	40	99.91
	5	[70, 70]	['id', 'marks', 'section']	30	99.28
	6	[120, 20]	['id', 'lang']	100	94.78
Cluster 2	7	[100, 100]	['tid', 'school']	30	94.6
	8	[110, 130]	['tid', 'school', 'teacher', 'salary']	150	71.64
	9	**[120, 110]**	**['tid', 'school', 'teacher', 'salary', 'address']**	**55**	**91.89**
	10	[130, 120]	['tid', 'address']	30	89.38
	11	[90, 120]	['tid', 'subject']	30	93.7
Cluster 3	**12**	**[30, 20]**	**['sid', 'staff_name']**	**150**	**82.18**

Nodes that are BOLD represent the Cluster Head Nodes

TABLE 2.1

have been obtained the cluster head now calculates the best suited node if any to be the optimized cluster head. In our node space *Node 0* being the initial cluster head selects its cluster members to be *Nodes [1, 2, 3, 4, 5, and 6]*. Now it goes on to find whether a better cluster head can be found among these nodes. After calculation *Node 0* doesn't find any such node so it remains the cluster head and now acts as a co-ordinator for communication in new cluster formed.

Then from among the remaining nodes the next cluster head *Node 8* is chosen as the initial cluster head which in turn choses *Nodes [7, 9, 10, and 11]* to be its members. Now *Node 8* tries to find a better cluster head than itself, and in the process finds that *Node 9* is indeed a better cluster head node according to the previous defined factors. *Node 8* then demotes itself as the cluster head and promotes *Node 9* to be the acting cluster head to the newly formed cluster. Now only *Node 12* remains and it forms a cluster with it being the sole cluster member as well as the cluster head. Table 2.1 shows the node space after the first set of clusters are formed.

Cluster No	**NODE SPACE AFTER COMMUNICATION**				
	Nodes	***Position***	***Tables***	***Communication Range***	***Energy***
Cluster 1	**0**	**[50, 50]**	**['id', 'roll', 'name']**	**150**	**40.27**
	1	[20, 30]	['id', 'roll']	30	98.83
	2	[100, 0]	['id', 'address']	30	95.5
	3	[30, 70]	['id', 'name']	30	99.28
	4	[40, 50]	['id', 'name', 'roll', 'marks', 'section']	40	99.82
	5	[70, 70]	['id', 'marks', 'section']	30	98.56
	6	[120, 20]	['id', 'lang']	100	94.78
Cluster 2	7	[100, 100]	['tid', 'school']	30	94.15
	8	[110, 130]	['tid', 'school', 'teacher', 'salary']	150	71.19
	9	**[120, 110]**	**['tid', 'school', 'teacher', 'salary', 'address']**	**55**	**76.88**
	10	[130, 120]	['tid', 'address']	30	89.19
	11	[90, 120]	['tid', 'subject']	30	92.8
Cluster 3	**12**	**[30, 20]**	**['sid', 'staff_name']**	**150**	**82.18**

Nodes that are BOLD represent the Cluster Head Nodes

TABLE 2.2

Now suppose at *Site 5* an update request occurs or suppose at *Site 5* data needs to be replicated to be transferred to other sites. Let the columns in question to be updated/replicated

be *tid, school and marks*. Now *Node 5* in our node space communicates with the cluster head of its cluster i.e. *Node 0* about this request. *Node 0* now relays this information to the sites where the changes are required namely *Node 4* and *Node 5* in its cluster space. Now *Node 0* being the cluster head broadcasts a signal to find the other heads present in the node space. The other cluster heads respond back namely *Node 9* and *Node 12*. *Node 0* relays the request to

Cluster No	NODE SPACE AFTER RE-CLUSTERING				
	Nodes	***Position***	***Tables***	***Communication Range***	***Energy***
Inactive	0	[50, 50]	['id', 'roll', 'name']	150	32.33
Cluster 1	**1**	**[20, 30]**	**['id', 'roll']**	**30**	**98.64**
Cluster 2	**3**	**[30, 70]**	**['id', 'name']**	**30**	**97.03**
Cluster 3	2	[100, 0]	['id', 'address']	30	94.78
	4	[40, 50]	['id', 'name', 'roll', 'marks', 'section']	40	93.07
	5	**[70, 70]**	**['id', 'marks', 'section']**	**30**	**92.62**
	6	[120, 20]	['id', 'lang']	100	78.49
Cluster 4	7	[100, 100]	['tid', 'school']	30	93.25
	8	[110, 130]	['tid', 'school', 'teacher', 'salary']	150	70.29
	9	**[120, 110]**	**['tid', 'school', 'teacher', 'salary', 'address']**	**55**	**46.48**
	10	[130, 120]	['tid', 'address']	30	88.83
	11	[90, 120]	['tid', 'subject']	30	90.99
Cluster 5	**12**	**[30, 20]**	**['sid', 'staff_name']**	**150**	**79.75**

Nodes that are BOLD represent the Cluster Head Nodes

TABLE 2.3

these nodes. *Nodes 9* and *Node 12* finds out whether the request applies to any of its nodes present in its respective cluster spaces and sends them signal accordingly to update their schemas. In *Node 9* cluster space the *Nodes [7, 8, 10, and 11*] needs to be updated and their cluster head sends them the signal to do so. The node space after such an update request is presented in Table 2.2

Now if any node's energy falls down below *threshold limit (35)* then it goes inactive. If a cluster head energy falls below threshold limit then re-clustering of the cluster space takes place. The node space after such a re-clustering is shown in Table 2.3. Due to an update request at *Site 5* to change schemas *tid, marks and school* done 3 times, re-clustering takes place after *Node 0's* energy falls below threshold limit. The node space after such a scenario is shown in Table 2.3.

From Table 3 we can see that the energy consumption overhead is significantly lower if the node space is being clustered versus it being non clustered, which further cements our statement that due to clustering of the node space, even though more nodes are being visited it is being done so in an efficient manner with most of the task being done by the cluster head. Therefore it yields significant power saving than a traditional system where nodes communicate with each other without implementing any smart logic in their intra node communication.

COMPARISON TABLE			
Non-Clustered Communication		Clustered Communication	
Communication From Nodes	**Distance**	**Communication From Nodes**	**Distance**
5 to 4	36.0555	0 to 5	28.2843
5 to 7	42.4264	0 to 4	10
5 to 8	72.111	0 to 9	92.1954
5 to 10	78.1025	9 to 7	22.3607
5 to 11	53.8516	9 to 8	22.3607
ENERGY CONSUMTION 15.56		9 to 10	14.1421
		9 to 11	31.6228
		ENERGY CONSUMPTION 10.43	

TABLE 3

Table 3 draws comparison between how a communication as shown in Table 2.2, would take place in the node space if it was clustered verses being non-clustered.

VI. Conclusion

We studied the different types of algorithms that are there in the field of Data Replication. We propose a new algorithm which takes into consideration the energy consumption in such a scenario. We are optimising the algorithm further such that the clustering is being done in a more efficient way, further reducing the energy consumption factor in this scenario.

Acknowledgment

We are grateful to our project guide Ms. Anuradha Banerjee for her guidance, inspiration and constructive suggestions that helped us in the successful completion of this project.