I J C T A, 10(8), 2017, pp. 495-499 ISSN: 0974-5572

A Survey: on the Incremental Repartitioning techniques to improve the scalability of the OLTP system

Anagha Bhunje* and Swati Ahirrao**

ABSTRACT

Cloud Computing is rising technology in today's computing platform. It also provides storage solutions to the enterprises. Online Transaction Processing systems (OLTP) need to be scalable and require fast response. The applications with large databases where new records are continuously added to the databases and which increases workload on the server. Scalability is required to meet the rising demand for the services offered. For improving the scalability, data partitioning is most commonly used technique. The Partitioning techniques are used to manage large databases and improve query response time. In large databases, transaction requires the data items which are stored across different nodes. This data is distributed in nature. It results in expensive growth in the number of transactions. The work-load aware partitioning approaches are used to reduce the number of transactions that are distributed in nature. The related data tuples are collected together based on workload analysis and which requires repartitioning of the database. In this paper, the survey of the Incremental repartitioning techniques is proposed.

Index Terms: OLTP (Online Transaction Processing), Distributed transactions, Incremental repartitioning. Scalability

1. INTRODUCTION

Cloud Computing is the powerful technology which enables the users to use the resources on demand. The performance of the enterprises is fast, smart and efficient with the use of cloud applications and big data. A large database is needed for storing data in order to meet heavy demands. Several billions of customers are handled on the internet. The companies need to handle large amount of data. Such huge amounts of data are generated every day. Such data are distributed across different machines. Single machine does not handle such huge amount of data. Relational database does not efficiently handle such data. Relational database has fixed schema. Therefore Scalability of the e-commerce application has become the challenging task.

NoSQL databases can be efficiently scaled out by using various partitioning techniques. Partitioning techniques are useful for distributing the data among many machines. Data is easily available across many machines. Response time of the server gets increased. D.J. Dewitt and J. Gray [2] introduce the partitioning techniques for scaling database workload. The commonly used partitioning techniques are round-robin, range and hash partitioning. Accessing data tuples from geographically distributed server affects the database scalability. Due to rapid growth in the requests, response time of server is slowed down. So scaling modern OLTP applications is the challenging task. These techniques does not consider the relation between the tuples and end up with the cluster of the uncorrelated data tuples on the same partition. Due to that, costs of the transactions are increased.

Graph structure is a way to express relationship between different objects in the form of vertices and edges. Several users are connected to each other with the help of social network. Social network contains different entities information. The entities contain information about personal details, friends information

* Symbiosis Institute of Technology, Pune, *Email: Anagha927@gmail.com*

** Symbiosis Institute of Technology, Pune, *Email: swatia@sitpune.edu.in*

etc. There exists at least one relationship between the entities. These entities are distributed among different servers. Due to that, numbers of the transactions are increases. As graph size increases, numbers of entities involved in the graph are increases.

The graph partitioning technique is most useful technique to reduce the communication cost among the nodes and ensure the load balancing. There are two types of the graph partitioning: 1) Offline graph partitioning, 2) Online graph partitioning. The graph partitioning techniques are distributed the data across different machines. The numbers of the transactions are increases with the use of existing partitioning techniques.

Due to the use of the existing partitioning techniques, the response time of the OLTP does not fast and immediate. To improve the response time of the server, Incremental repartitioning technique will be the useful technique. So In this paper, the survey of the incremental repartitioning techniques is proposed. Existing partitioning techniques does not consider the relation among the attributes. Numbers of the partitions are increases if the database size gets increased. Some amount of the data gets lost due to load imbalance on the server.

In incremental repartitioning technique, related items are gathered together on one server. So, Load on the server is equally balanced. Number of the transactions are reduces.

2. RELATED WORK

A.Quamar et al. [2] introduces technique for incremental repartitioning of the database. The data is partitioned across the physical partitions. Hyper graph representation technique is used for representing workload, where nodes are the data items and each query is translated to hyper edge over the nodes. The k-way balanced min-cut partitioning technique is used to get balanced partitions of the database. The goal is to store each data item (node in the graph) on a subset of machines (partitions), by observing storage capacity requirement of the partitions. Hyper graph technique is used to identify commonly accessed items in previous transactions.

This paper introduces data replicated system. Replication is a technique of storing exact copy of information at some other place in case of fault tolerance. There are two approaches of replication. 1) Active replication, 2) Aggressive Replication. Active replication suffers from the problem of maintaining ACID properties. It does not balance the load properly. Aggressive workload aware replication technique is used in this paper to avoid disadvantage of the active replication. In this technique, data availability is proportional to the workload requirements.

Curino et al. [3] describes the workload aware approach for achieving scalability with the help of graph partitioning. In graph representation technique, Graph structure represents the entire transaction workload. The node indicate the individual item in the transaction. Edges that connect the nodes that are used in the same transactions. Tuple level replication is extension to the graph representation. In this technique, star shaped configuration is used for the representation of replication. In graph partitioning, the data items which are used by the transactions are located on same partition

Miguel Liroz-Gistau et al.[4] introduces new dynamic partitioning algorithm Dynpart which handles continuously growing databases. It is useful for the data partitioning in many applications where records are added frequently to the databases. The data set is divided into number of fragments.

This algorithm work on a set of data items instead of single data item. This algorithm helps in selecting best partition to place new data items. It selects the partition based on the closed relation between the attributes. If there are several fragments that have highest affinity (close relation) then smallest fragment is selected in order to keep the partitioning balanced. The total size of the database does not considered while evaluting the performance of the algorithm.

As data volumes are growing continuously, Data distribution is a big problem in order to achieve scalability and availability. Xiaoyan Wang et al.[5] explains the new strategy ADDS (Automatic Data Distribution Solution) for OLTP applications. This technique helps in solving the problem of data distribution based on data and workload analysis. It distributes both static and incremental data. For data distribution, one triangular model DaWN (Stands for Data, Workload and Nodes) is proposed. Data, Workload, Nodes are represented as vertices of the triangle. ADDS contains two phases: 1) Data partitioning, 2) Data placement.

Data partitioning is used to split the data into fragments and then cluster the fragments into partitions based on the affinity scores. In data placement phase, each partition is assigned to the single node.Alexandru Turcu et al [6]developed a new methodology for automatic data partitioning in Granola based distributed transactions. Online transaction processing system requires fast response. Such system many times suffers from the problem of weak consistency. Granola model handles the independent transactions in order to keep the transactions serializable.

This paper extends the Schism [3] to support the independent distributed transactions. Granola model tries to execute all transactions in the single partition. The locking mechanism is used by the Coordinated distributed transactions to perform two phase commitment process. This paper develops independent distributed transactions which enable atomic commitment across a set of transactions without agreement and locking mechanism.

1) It collects data dependency information to avoid dependencies among the partitions.
2) It finds which operation to be performed within single block.
3) It helps in recording workload traces.

The graph representation technique is used to represent workload into graph in which nodes indicate the attributes and edges indicate the transactions. The METIS graph partitioning algorithm is used to partition the graph into fine grained partitions.

Francisco Cruz et al. [7] introduce an automated table splitting mechanism. Recent data stores partition the region in such a way to distribute load across the nodes. In this paper, it finds the good splitting point. A good splitting point is one that splits the region into two regions with similar load. This paper introduces new algorithm for finding good splitting point.

Swati Ahirrao et al. presents [8] presents the dynamic workload aware partitioning. In this partitioning, transactional logs are used for analyzing workload. With the help of these logs, frequently occurred data item sets are gathered. Based on this, partitions are formed to balance the load among all partitions. In this paper, first frequently used data items are kept on one domain on SimpleDB and response time is measured. After that, those data items are kept on two different domains and response time is measured. After comparing the response time of transactions in both cases, it is analyzed that response time is slow when data items are kept together on one domain.

The number of tuples in the same transaction. The graph partitioning algorithm is applied on the graph in order to obtain equal load on the server. The load on the server is balanced in such a way that large number of the adjacent edges are on same partition. Decision tree classifier provides range predicate partitioning. Graph based schemes is used for improving response time and latency.

The number of tuples in the same transaction. The graph partitioning algorithm is applied on the graph in order to obtain equal load on the server. The load on the server is balanced in such a way that large number of the adjacent edges are on same partition. Decision tree classifier provides range predicate partitioning. Graph based schemes is used for improving response time and latency.

Chao-Wei Ou [9] explains solution to the incremental graph partitioning problem. Incremental graph partitioning means number of the nodes are added or removed at any instant of time. Graph partitioning

needs to be done in parallel way . Partitioning technique helps in balancing the load among the partitions. Computational load on the node grows continuously. By observing storage capacity of the node, assigning each partition to one processor becomes difficult task.

Steps:

1. Assign new tuples to one of the partitions based on the affinity score.
2. Assign each tuple of the given partition to another partition it is close to.
3. Balance the load.

Reduction in the number of the distributed transactions in shared nothing distributed database is difficult task for the transactional workloads. Nowadays, there is a tremendous growth in the data volumes. Incremental repartitioning [10] is tries to reduce the number of distributed transactions. It tries to convert the distributed transactions into non-distributed transactions. Graph or Hyper graph representation technique is used to represent the workload and these graphs or hyper graph are repartitioned to generate the data migrations. Transaction classification technique is used to identify distributed and non-distributed transactions. The Transactions which contain at least two items are placed on two different servers are referred as distributed transactions. Nondistributed transactions are further divided into Movable and Non-movable transactions. This technique helps in reducing the size of the workload.

Clustering is the technique of grouping the objects based on the similarity and between objects. Partitioning techniques form clusters to place the data. Hierarchical Incremental clustering algorithm [11] is used to repartition the attribute set when attributed are added to the database continuously. It starts partitioning into clusters that was established previously.

Rajkumar Buyya [12] introduces workload aware incremental repartitioning technique for cloud applications. The Cloud applications require fast response while executing their transactions. The data tuples are added dynamically to the server. Load on the server increases. Numbers of the transactions are increases. Workload aware incremental repartitioning technique is used to reduce the communication cost and load among the nodes. Entire workload is represented as hyper graph. K-way min cut clustering algorithm is used to balance the load among the partitions.

The author Andrew Pavlo[13] introduced a new approach for automatically partitioning a database in a shared nothing , parallel DBMS . Horticulture is the automatic design tool which helps in selecting physical layout for DBMS. The new database design is considered the amount of the data and transactions assigned to the single partition. Horticulture analyses a database schema, the structure of the applications stored procedures, and a sample transaction workload, and then automatically generates partitioning strategies that minimizes distribution overhead while balancing access skew. Horticulture makes use of Large-neighborhood search (LNS). LNS compares potential solutions with a cost model that analyses the DBMS will perform using a particular design for the sample workload trace without needing to actually deploy the database. Existing partitioning approaches does not eûectively handle the load . It cannot handle the dynamic workload change. So incremental repartitioning technique is used for reducing number of the distributed transactions.

3. CONCLUSION

Incremental repartitioning technique is useful in improving scalability of the OLTP systems. This paper includes the detailed survey of the incremental repartitioning techniques. Incremental repartitioning technique handles such applications where data items are continuously added to the database. It helps in gathering most frequently accessed items together. These techniques are also improve the response time of the server. The clustering algorithm tries to cluster commonly used items together on one partition. So the number of

the transactions are reduces .The main goal of the incremental repartitioning technique is to convert distributed transactions into non-distributed transaction.

REFERENCES

[1] D.J.Dewitt, J.Gray, "Parallel database systems: the future of high performance database systems", ACM (1992).

[2] A. Quamar, K.A. Kumar, A. Deshpande, "SWORD: Scalable Workload-aware data placement for transactional workloads", Proceedings of the 16th International conference on Extending Database technology, ACM (2013).

[3] C. Curino, E. Jones, Y. Zhang, S. Madden, "Schism: a workload-driven approach to database replication and partitioning", Proc.VLDB Endow.3 (1-2) (2010).

[4] Miguel Liroz-Gistau, Reza Akbarinia, Esther Pacitti, Fabio Porto, Patrickvalduriez, "Dynamic workload-based partitioning for large scale databases", lirmm-00748549, version-1-5 (Nov 2012).

[5] Xiaoyan Wang, Xu Fan, Jinchuan Chen, Xiaoyong Du, "Automatic Data Distribution in Large-scale OLTP applications", International Journal Of Databases theory and Applications,vol.7,No.4(2014).

[6] Alexandru Turcu,Roberto Palmieri,Binoy Ravindra, Sachin Hirve, "Automated Data Partitioning for Highly scalable and strongly consistent transactions", IEEE transactions on parallel and distributed systems,Vol-27, January (2016).

[7] Francisco Cruz, Francisco Maia, Rui Oliveira, Ricardo Vilaca, "Workload aware table splitting for NoSQL", ACM,14 March (2014).

[8] Swati Ahirrao, Rajesh Ingle, Dynamic Workload Aware Partitioning in OLTP cloud data store", JATIT LLS, Vol. 60 No. 1, (Feb 2014).

[9] Chao-Wei Ou, Sanjay Ranka, "Parallel Incremental Graph Partitioning", IEEE Transactions, Vol. 8, No. 8, (August 1997).

[10] Joarder Mohammad Mustafa kamal, Manzur Murshed, Mohamed Medhat Gaber, "Progressive Data stream mining and transaction classiûcation for workload aware incremental database repartitioning", IEEE, (2014).

[11] Joarder Mohammad Mustafa kamal, Manzur Murshed, Rajkumar Buyya, Workloadaware Incremental Repartitioning Of Shared nothing Distributed databases for scalable cloud applications, IEEE, (2014).

[12] 13.Pavlo, Andrew, Carlo Curino, and Stanley Zdonik, "Skew-aware automatic database partitioning in shared-nothing, parallel OLTP systems", Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, No.149, 2012.

[13] 14. Kamal, Joarder, Manzur Murshed, and Rajkumar Buyya. "Workload-aware incremental repartitioning of shared-nothing distributed databases for scalable OLTP applications", Future Generation Computer Systems 56 (2016).