I J C T A, 10(8), 2017, pp. 501-507 ISSN: 0974-5572

Survey: A Wide Survey on Graph Partitioning Algorithms

Arvind Sahu*, Swati Ahirrao*, Maya Shelke* and Shraddha Phansalkar*

ABSTRACT

Today, millions of the distributed users concurrently access the web applications and websites of the enterprises with the expectation of high availability and reliability. The workload is to be represented in the form of the graph, to take the advantage of the relationships between them. Graph Based workload driven partitioning system framework, is to improve the scalability of NoSQL Database, to improve the transaction response time and latency using graph partitioning and decision tree techniques. Graph partitioning is an important technique to make easy representation and processing of Online Transaction Processing (OLTP) workload for millions of concurrent users. Graph partitioning partition the data and keep related data into a single partition. If the partitions are not formed appropriately, the result becomes expensive in terms of distributed transactions. A good graph partitioning technique is required for efficient partitioning because it has a direct effect on communication overhead and load balancing. Graph partitioning algorithm is applied to create partitions of a graph. The result of graph partitioning becomes a highly balanced partition of graphs. The overall aim of graph partitioning is to make database more scalable with the reduced expensive distributed transaction

Keywords: OLTP workload, Graph based data partitioning, Distributed DB, Distributed transaction, NoSQL DB.

1. INTRODUCTION

World Wide Web (WWW) contains about 70 billion web pages and more than trillions of different URLs. To represent these graphs is used because of relativity between nodes. The biggest social networking website is Facebook. It contains 1000 million nodes and over 100 billion links. It is very difficult to process large data sets by using traditional database management tools. Here NOSQL databases like MongoDB, CouchDB etc. are base storage units of Big Data. NOSQL is popular data storage and data retrieval because it supports better data availability, scalability, and faster data access and process as compared to traditional relational database management systems. A graph is a common data structure. General relations and complicated relations can be represented in graph. Graphic representation is better than to linear arrays and tree. Realistic scenes can be presented by the use of graph.

A partition is a division of a logical database into distinct independent parts. Graph partitioning is a fundamental technique for Parallelization. The objective of graph partitioning is to divide the given graph into smaller parts also call vertex subsets, such that they have a minimal number of edges cut and roughly equal computational loads. The aim of Graph partitioning algorithm is to fulfill all requests locally, that will reduce communication among graph partitions and the result becomes less distributed transaction. Database partitioning is normally done for performance, availability reasons, manageability, and for load balancing. Partitioning helps in reducing the total cost of the storage of large amounts of data. There are some existing data partitioning schemes as List, Range and Hash partitioning.

Graph partitioning algorithms is used to find a related data partition so that the number of distributed transactions becomes less and which increase scalability, manage- ability and availability. Graph partitioning

* Symbiosis Institute of Technology, Pune. Computer Science Department, *Emails: Sahuarvind1993@gmail.com, swatia@sitpune.edu.in, mayas@sitpune.edu.in, shraddhap@sitpune.edu.in.*

is applicable in many fields, like scientific computing, workload driven, VLSI design and in the operating system for task scheduling, online transactions, web analysis and many more areas. The data has to be partitioned so that the load of accessing data can be easily distributed and the formed partitions among distributed system can be kept. In this paper, all possible Graph partitioning techniques are studied.

Graph partitioning has two main phases, first develop the graph from transaction workload. Where tuples which are accessed in the same transaction are connected by edges and database tuples are represented as nodes. Graph partitioning technique is used to find min-cut balanced partition. Second, the decision tree classifier is used to predict a range formed by the graph partitioning algorithm.

2. RELATED WORK

Curino C et al. [1], have proposed Schism technique for database replication and partitioning for OLTP databases. Schism is a data driven graph partitioning approach designed to improve scalability of distributed databases. Schism represents the workload in the form of a graph. To reduce the effect of distributed transactions, it applies k-way [15] graph partitioning algorithm which also increases the throughput.

There are four partitioning phases in schism. First, Graph Representation phase converts the incoming load in the form of a graph. Node represents an object and Edge represents a transaction. Weight can be assigned on either number of transaction or data size. Second, In Graph Partitioning phase previously connected graph is partitioned using a standard METIS [2] graph partitioning algorithm. Each and every tuple is assigned to one partition, and then each partition is assigned to one physical node that is vertex. K-way partitioning is used in the initial phase of METIS for finding refined partition. Third, In Explanation phase it captures the tuple partition mapping as they were produced in partitioning phase. Decision tree classifier is used for developing some rules. Fourth, Validation phase is used for comparing results of the cost of per tuple partitioning and range predicates partitioning against hash partitioning by considering the total number of distributed transactions as a metric. The graph generated by Schism is extremely huge and usually does not deal with progressive workload deviation and re-partitioning. In this research paper, they propose a number of heuristics, including sampling and record clustering, which reduce graph complexity and optimize performance.

The advantages of Schism are modest runtime, excellent partitioning performance, ease of integration into existing databases, impressive n-to-n relationship representation, and no restriction on schema layout, a fine-grained and consolidate approach for replication and partitioning of social networking data. Apart from advantages, there are also some limitations of Schism are, it does not deal with changes in workload, Schism uses aggressive replication (copy data nodes with a less write/update frequency).

GEORGE KARYPIS et al. [2], have proposed METIS for Graph Partitioning. METIS follows multilevel graph partitioning technique. There are three phases in multilevel graph partitioning, firstly graph coarsening is to be partitioned the graph developed by incoming workload also called as projected partition, then initial partitioning is to further partitioning the small graphs for refinement and the result becomes refined partition which will be balanced partition, and third un-coarsening is just opposite of coarsening for rebuilding the final graph. For refined partitioning k-way algorithm in initial partitioning phase is used.

METIS is one of the best open source graph partitioning techniques. METIS provides two different interfaces. First, simple interface is provided via the programs p-metis, k-metis, and o- metis for users that want to use METIS to either partition or order a sparse matrix. Second, advanced interface is for the users that want to experiment with the different parameters of the algorithm. METIS is highly optimized multi-level algorithm. They proved that the METIS multilevel recursive bisection algorithms are 11 to 41 times faster as compared to multilevel spectral bisection. METIS multilevel k-way partitioning algorithms are 41 to161 times faster than multilevel spectral bisection. METIS is a better edge cut graph partitioning tool.

METIS have many advantages like to provide quality partitions, comparatively fast, reduce fill orderings and minimizes the amount of memory it requires, by dynamically allocating. Along with advantages, there are some limitations of METIS like hard to estimate the exact amount of memory required by METIS.

Tatarowicz AL, et al. [3], have proposed a Fine- Grained Partitioning for distributed database. In proposing partitioning, related individual tuples are co-located together in the same partition. In Lookup tables manual partitioning can be implemented. To maintain a fine- grained partitioning they propose Lookup table because it is needed to store the location of each tuple. The lookup table store location of each tuple, so it is also known as metadata table. Many-to- many relationship increases the distributed transactions, which is hard to partition. They present a design that is efficiently able to store very large tables. It maintains that, as the database is modified. It increases the scalability of database workloads. That workload is difficult to partition like Wikipedia, Facebook, Twitter, and TPC-E.

Fine grained partitioning provides about 41 percent to 301 percent better throughput for workloads than other partitioning. Set of techniques is introduced to store efficiently and compress lookup tables. Manage updates, inserts, and delete operation are also described for performing any operation on Lookup table. Data within Lookup table contain either many-to-many relationships or queries. Data can be filtered using different keys which are unique like id or title. For example, One's email can be accessed based on unique key like email-id. Lookup table should be stored compactly in RAM. Queries on different attributes should be broadcasted between partitions.

Fine-Grained Partitioning has many advantages like the number of distributed transactions can be reduced by well assignment of tuples to each partition, for different workload, it can improve the performance, to reduce the main memory consumption, the same lookup table is reused for cached the database lookup tables in main memory. There are some limitations of Fine Grained Partitioning are routing (Lookup) tables very large and expensive in maintaining, the initial cost of partitioning can be very high, not clear how to handle and process newly inserted tuples.

Quamar A et al. [4], have proposed Scalable Workload-Aware Data partitioning for Transactional data processing. They represent workload in the form of the hypergraph. Hypergraph reduces the overheads of partitioning by hyperedge. SWORD techniques basically reduce the overheads. Overheads happen at the time of data placement or query execution at runtime. The incremental data re-partitioning technique is used to deal with the workload changes. They use the active replication technique to increase the availability. The incremental data re-partitioning can use to modify a set of data.

SWORD functions in three steps. First, In Data Partitioning and Placement the data is horizontally partitioned. These modules are in use for taking the decision of data placement decisions and data replication. Second, workload change is monitored by incremental re-partition. Incremental re-partitioning also maintains statistics. To identify when the current partition triggers data onto another partition and suboptimal to deal with workload change input is provided to the incremental re-partition. Third, transaction processing is interface module for submitting transaction and to receive the results. Transaction manager provides ACID property by two phases commit protocol. They have devised techniques to minimize the cost and maintenance overheads of graph partitioning. Less number of distributed transactions, the act of tolerating fault, increased accessibility, and load balancing can be accessed by using active replication.

Advantages of SWORD are used to minimize the cost of distributed transactions. SWORD introduces workload-aware replication mechanism for better data placement. SWORD reduces the overhead at the time of query execution. SWORD effectively represents and compresses data. SWORD generates the less number of partitions.

Lu Wang et al. [5], have proposed a Multi-level Label Propagation (MLP) method for partitioning of the graph. It is very hard to partition the graph with billions of nodes. Here nodes representing to the web

users. Efficient partitioning makes easy load balancing and less communication overhead. MLP is a big challenge because the data cannot be organized in arbitrary ways to enhance the performance of the partition by using partitioning algorithm. They prove that, billion-node graphs can be partitioned within a particular time on a distributed system. The quality of generating partitions by MLP approach is compared to the small size graph. The paper aims to represent web data in the form of a graph. Then the developed graph is broken into multiple small graphs. To evaluate quality of the formed partitions it is compared to METIS.

Multilevel graph partitioning happens in three steps, first graph coarsening second, initial partitioning and third un-coarsening. Firstly graph coarsening is to be partitioned the graph developed by incoming workload also called as projected partition, then initial partitioning is to further partitioning the small graphs for refinement and the result becomes refined partition which will be balanced partition, and third un-coarsening has been just opposite of coarsening for rebuilding the final graph. For refined partitioning k-way algorithm is used in initial partitioning phase.

Advantages of MLP are semantic awareness and efficiency, easily parallelized, more effective. MLP is scalable partitioning which can scale up to billions of nodes. MLP is able to find a good partition with respect to memory and time. Along with advantages, there are some limitations like MLP algorithm which are not designed for general-purpose.

Chris H.Q. Ding et al. [6], have proposed a Min-max Cut algorithm. The aim of Min-Max Cut (or M-cut) is to reduce the number of similar nodes between different clusters and increase number of similar nodes within a cluster. It means similar nodes between two subgraphs should be less and the similar nodes within each sub-graph should be higher.

M-cut works in three steps, first measure the Fiedler vector, should have sorted Nodal values to gather Fiedler order. Second, we find minimum M-cut point. On partitioning a number of edge-cut should be optimally minimized. Third, Linkage based refinement. It is used to improve the partitioning quality. Refinement algorithms are used to separate nodes into different partitions. Those different partitions should have the same number of nodes that is also called balanced partition. Along with that Min-cut shows that linear search based on linkage differential gives better result as compared to the Fiedler vector. Min-max (M-cut) cut is a better way of partitioning than the normalized cut and ratio cut. The main advantage of Min-max cut is that, it improves the clustering accuracy and gives an optimal solution.

Tae-Young Choe et al. [7], have proposed k-way Graph Partitioning Algorithm to reduce the size of the graph. K-way graph partitioning is based on clustering. The recursive spectral bisection method is used for clustering small graph in a k-way. The recursive spectral bisection method reduces the size of the graph by the breakdown edges and vertices. Balancing constraints should maintain to all partitions of graphs. K-way is working within initial phase of Multi Level Partitioning for recursive partitioning.

There are three steps for partitioning a graph in k-way. First, vertex connectivity to cluster the graph is considered in clustering step. The proposed algorithm divides all the vertex clusters of the same size. Second, the clusters that are generated during clustering are balanced in Balancing step. Third In Refinement phase, firstly balanced partitions are found and Refinement on that partition is applied in Refinement phase. In Refinement, the subsets of a node are swapped from one partition to another partition. Refinement reduces the cut-set size. Swapping subsets should have the same number of nodes. KL-Refinement algorithm and Sanchis Refinement algorithm are good for Refinement. Partitioning by using k-way algorithm generates balanced partitions. These partitions are highly balanced as compared to the Multilevel Label Partitions (MLP). A small number of cut-set in K-way makes it an effective partitioning algorithm. As the transaction between two different cluster nodes increases the number of edge cut also increases simultaneously.

The main advantage of k- way is its partitioning performance in terms of Time; k-way is a better option for partitioning small graphs. Limitation, we can use of k-way algorithm to partition of small graphs.

Lyu-Wei Wang et al. [8], have introduced BiFennel for Fast Bipartite Graph Partitioning Algorithm. BiFennel is inspired by Fennel [12] and BiCut [13]. BiFennel minimizes the time taken to process the graph and load that processed graph onto the network by maintaining a less number of vertex replication factor. BiFennel also maintains work balance. PowerGraph is used to implement BiFennel. PowerGraph cut vertices to partition a graph.

BiFennel algorithm has four steps. Firstly, the whole transaction data in the form of a graph is represented or loaded and is stored in the neighbor list. The list contains each vertex's neighbors. Now to find master node and replicate vertices graphs, list and tree are constructed and other data structure which are needed. In PowerGraph the finalized function is called to represent the final output in the form of statistics.

BiFennel provides good communication cost. BiFennel has 30 to 50 percent improvement on communication cost. BiFennel also provide good runtime as comparable to Aweto. BiCut and Aweto are also vertex cut algorithms, implemented on PowerGraph. The main advantage of BiFennel has overall good performance, and minimize communication cost. BiFennel requires less runtime.

Along with advantages, there are some limitations like speed and memory allocation. BiFennel has rapid speed and less memory.

Dominique LaSalle et al. [9], introduce Parallel Hill-Climbing algorithm. Parallel Hill-Climbing is a Graph Partitioning algorithm. This algorithm is used for share memory for refinement algorithm parallel. Hill-climbing is imaginary move on different vertices to find new local minima. The refinement algorithm produces high-quality partitioning and capabilities to partition local minima. Hill-Scanning is the refinement algorithm for refining k-way partition in parallel and processing units can be scalable. In hill vertices will be moving from one transaction to another.

Initially Hill will be empty, so firstly root vertex into the order queue is added. Then vertices will be extracted from the order queue to the hill. Then entire hill gain is calculated, it must be positive for return hill. If it is positive, then return hill and exit loop. Otherwise the neighbors are added in the same partition. At a maximum size of hill it should produce results as a positive gain. If the resultant output of a hill is not in terms of positive gain, it is discarded.

Hill-Scanning is better than other parallel refinement algorithm because it is much faster and provides better quality. Hill-Scanning can spread to local minima. Advantages of proposed algorithm Hill-Scanning can be efficiently parallelized to achieve normal performance.

Maria Predari et al. [10], introduce Coupling-Aware Graph Partitioning Algorithm. They have proposed a new shared memory parallel algorithm. It is used to produce high-quality partitions as compared to the k-way partitioning algorithm. Co-partitioning allows scaling more in term of the number of processing unit. In Coupling-aware partitioning there is no need to partition codes independently or separately. The idea is to reuse available codes through a coupling framework without combining them into a standalone application. There is AWARE and PROJREPART called as co-partitioning algorithms.

AWARE method is divided into 3 main steps, First symmetrically applied to graph using the restriction operator for computing the coupled subgraphs based on the inter edges. Second, partitions, each subgraph independently. Third, using the extension operator obtained partitions stretched to the global graphs. PROJREPART works exactly same as AWARE method. The only difference is, it replaces the naive partition of coupled subgraph by the projection & repartition steps. PROJREPART is to improve the communication scheme during the coupling phase by minimizing both the number of messages and the amount of data exchanged.

The advantages of AWARE and PROJREPART algorithms succeed to balance the computational load and reduce the coupling communications costs. The algorithm provides better global graph edge cut.

Limitations, algorithm should be implemented in parallel and should be extended in a proper order to manage more complex applications.

James D. McCaffrey et al. [11], have proposed Simulated Bee Colony Algorithm to partition the graph. SBC is totally following the foraging behavior of honey bees. In SBC algorithm the partial result can be reused further. So it is compatible in a scenario where partition result can be reused. SBC algorithm is tested with respect to twelve benchmarks.

Foraging bees typically have three roles and on the basis of these roles Foraging bees find a solution. First Active foraging bees start to travel from food source to food destination. At the food destination, they gather food come back to the hive. Second Scout foraging type bees firstly find out the best food source enclosed to the hive. Scout bees always choose attractive food sources. Active and Scout bees perform waggle dance. Through the waggle dance they tell to the incoming audience about destination food quality, returned from Active and Scot. Active and Scot return the quality and location of destination food to the hive. On the basis of review of Active and Scot, waggle dance will be performed. Third Inactive foraging bees wait until Active and Scouts tell to hive about the quality. Waggle dance is a way to convey information about the source of food for inactive foragers. The number of cutset will totally depend on destination food quality. Inactive would minimize the number of cuts.

Advantages of proposed algorithm produce very high quality of partition. SBC is used when the quality is more important than performance like online road map applications. Limitation, SBC algorithms is not suitable for real-time performance like parallel processing. SBC performance is low.

3. CONCLUSION

This paper discusses all possible Graph partitioning algorithms used in data partitioning for NoSQL database. Graph partitioning is one of the promising techniques used to reduce the number of Distributed transaction issue by using the min-cut edge. In every algorithm author has proposed a different strategy to deal with different problems like workload, distributed transaction, load-balancing, scalability and many more. Multilevel partitioning and METIS are good graph partitioning algorithm, which are mostly used.

REFERENCES

[1] Curino C, Jones E, Zhang Y, Madden S. Schism: a workload-driven approach to database replication and partitioning. Proceedings of the VLDB Endowment. 2010 Sep 1;3(1-2):48-57.

[2] GEORGE KARYPIS AND VIPIN KUMAR.METIS Unstructured Graph Partitioning and Sparse Matrix Ordering. Army High Performance Computing Research Center under the auspices of the Department of the Army.

[3] Tatarowicz AL, Curino C, Jones EP, Madden S. Lookup tables: Fine-grained partitioning for distributed databases. In Data Engineering (ICDE), 2012 IEEE 28th International Conference on 2012 Apr 1 (pp. 102-113). IEEE. [13].

[4] Quamar A, Kumar KA, Deshpande A. SWORD: scalable workload-aware data placement for transactional workloads. In Proceedings of the 16th International Conference on Extending Database Technology 2013 Mar 18 (pp. 430-441). ACM.

[5] LuWang, Yanghua Xiao, Bin Shao, HaixunWang ICDE. How to Partition a Billion-Node Graph, semantic schola 2014.

[6] Chris H.Q. Ding, Xiaofeng He, Hongyuan Zhab, Ming Gu, Horst D. Simon . A Min-max Cult Algorithm for Graph Partitioning and Data Clustering. 2001 IEEE.

[7] Tae-Young Choe. K-way Graph Partitioning Algorithm Based on Clustering by Eigen Vector.

[8] Lyu-Wei Wang, Shih-Chang Chen, Wenguang Chen, Hung-Chang Hsiao3 and Yeh-ChingChung. BiFennel: Fast Bipartite Graph Partitioning Algorithm for Big Data. 2015 IEEE.

[9] Dominique LaSalle and George Karypis. A Parallel Hill-Climbing Re_nement Algorithmfor Graph Partitioning. 2016 IEEE.

[10] Maria Predari and Aurelien Esnard. Coupling-Aware Graph Partitioning Algorithms:Preliminary Study.IEEE 2014.

[11] McCa_rey, James D. “Generation of pairwise test sets using a simulated bee colony algorithm.” Information Reuse and Integration, 2009. IRI’09. IEEE International Conference on. IEEE, 2009.

[12] C. Tsourakakis, C. Gkantsidis, B. Radunovic, and M. Vojnovic, "Fennel: Streaming graph partitioning for massive scale graphs," In Proceedings of the 7th ACM International Conference on Web search and data mining, pp. 333-342, 2014.

[13] R. Chen, J. Shi, B. Zang, and H. Guan, "Bipartite-oriented distributed graph partitioning for big learning." In Proceedings of 5th Asia-Paci_c Workshop on Systems, pp. 14, 2014.

[14] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin, "PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs," OSDI. Vol. 12. No. 1. 2012.

[15] Karypis G, Kumar V. multi level k-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed computing.1998 Jan 10;48(1):96-129.

[16] Turcu, Alexandru, et al. "Automated data partitioning for highly scalable and strongly consistent transactions." *IEEE Transactions on Parallel and Distributed Systems* 27.1 (2016): 106-118.

[17] Kanase Shivanjali, and Swati Ahirrao. "GRAPH BASED WORKLOAD DRIVEN PARTITIONING SYSTEM FOR NOSQL DATABASE." *Journal of Theoretical and Applied Information Technology* 89.1 (2016): 85.

[18] Karypis, George, and Vipin Kumar. "Multilevel graph partitioning schemes." *ICPP (3)*. 1995.

[19] http://www.worldwidewebsize.com/.

[20] http://www.facebook.com/press/info.php?statistics.