A Survey on Shortest Path Estimation using RDBMS over graph

Priyanka S.Naphade, Y.V.Dongre

Department of Computer Engineering
Vishwakarma Institute of Information Technology Pune, Maharashtra, India
Email: priyankanaphade2@gmail.com, yashwant.dongre@gmail.com

Abstract - Recently with the advancement of social networking sites vast amount of data emerges out, which is usually in the form of graph data. Many graph search queries are in use to access and retrieve this data. Shortest path computation is required in many applications for their faster execution and can be considered as an important problem in such applications. This paper presents a survey of relational database system for graph search queries such as shortest path computing. Also, survey shows relational database based operations of graphs.

Keywords: Relational database, graph, shortest path.

I. INTRODUCTION

Nowadays, Graph data emerges at a very fast rate. Such huge amount of data should be managed efficiently which abandon many challenges to graph management. Graphs provide a mechanism to represent relationships between objects, where some pairs of objects are connected by links. Graphs have many applications such as in social networking, for transportation and phone networking. Also graph is used for linking of web pages. For such applications graphs are in large sizes. Such huge graphs are difficult to manage and continue to grow exponentially. Data generated with the help of these areas is known as graph data and in order to execute these applications operations need to be done on data such as graph search. Facebook makes use of graph search for connectivity purposes. Graph search suffers from many challenges such as how to rank the results etc. Accordingly, to develop and analyze applications with high amount of graph data efficiently, various graph algorithms are developed such as the shortest path finding, minimal spanning tree. The shortest path discovery is an important graph search query and plays crucial role in many Applications such as social networking in which connections among individual is important.

Many algorithms are well known that solve shortest path problem such as Dijkstra’s algorithm, Prim’s algorithm to find minimum spanning tree. As various areas provides huge graph data consisting of more than thousands of nodes All these areas and applications makes graph search queries difficult to manage. There are many ways which can support graph search queries. Nowadays, network sizes are increasing rapidly, and the networks cannot fit into the memory. So in-memory approaches require to be addressed. Thus, memory based algorithms for shortest path discovery have deterioration in performance or cannot work efficiently in the large-scale networks. There are many disk based methods to support large-scale graphs. A single computer based approach proceeds by dividing a large graph into sub graphs that fit into memory and loads them into the memory to process partially.

The other approach which is used for big data is a Map Reduce based distributed processing approach. This is used to manage huge graphs in distributed manner. However, because of its schema less representation, results in many challenges.

Relational database has proved itself effective for graph management from several years. Relational databases are more mature, stable and can handle most complex data types efficiently. It provides many functionality that is used for managing graph data. Indeed, RDB provides a basic support to the graph storage and the flexible access to the graph. In addition, RDB has already shown its capabilities in managing other complex data types and supporting novel applications. However, it is a challenging task to translate the graph search methods and improve the performance in the RDB context since relational operation and graph operation are different.

The remaining paper is organized as below. In section 2 we explained the related work to graph search, graph operations and relational database based graph operations. In section 3 we presented the basic Framework which is used to find shortest path in relational context.

II. RELATED WORK

A. Work Related to Graph Search:
Dijkstra’s algorithm is eminent online algorithm [19] to solve the single-source shortest path problem. For graph search many existing approaches use selection, expansion and merging processes.

D. Wagner et al. [14] proposed many techniques for improving performance of computation of shortest path, such as Bidirectional search which is an extension to Dijkstra’s algorithm. Goal Directed search that depends
upon available space and time required to perform some preliminary operations. Bidirectional search proceeds by searching path in both directions i.e. forward and backward and path is calculated when forward search meets backward search. Goal Directed techniques proceed by considering priorities of the nodes and changing that priorities by adding heuristics to cost of the node. Another approach is Hierarchical approach in which a hierarchical structure is made by adding some additional edges to the graph. By building this structure it makes use of sub graphs and computes the path with minimum distances.

Many approaches take into account various indexes such as Landmark index. M. Potamias et al. [8] provided solution for shortest path in a large network and used Landmarks index which are computed using Random, Degree, Centrality strategies based on certain conditions and division of graphs and shown effective when used in networks when results need to be accurate and appear in less time. E. Cohen et al. [18] used 2-hops index which considers a few nodes which then considered as landmarks for optimization purpose but time required is too large for practical limits.

Fang Wei [4] proposed an approach for shortest path computation based on indexing and tree decomposition method which uses bottom-up approach for finding shortest path in linear time and uses tree decomposition algorithms that can be applied to graphs with large size which fit into main memory.

Many approaches have been proposed which work with external graph operations. D. Hutchinson et al. [17] proposed an approach for storing large planer graphs which are huge to store into memory. When Graph sizes increases tremendously then it not possible for internal memory to store it. PDM, An external shortest path index is proposed on planar graphs. This approach uses a data structure for graphs that are not able to store into the main memory and handle graph data efficiently. C. Aggarwal et al. [7] proposed an Edge sampling based approach to create compressed representations of the graphs and Connectivity index for such graphs. Large graphs are compressed and stored in main memory which is then arranged in an index structure to work with disk.

J. Dean et al. [16] proposed a framework used for large graph for distributed processing for big data. It poses disadvantages in terms of cost and does not provide support for dynamic updatation of graphs.

B. Work related to showing RDBMS capabilities:

Relational database have been proved to support several data types over years. F. Tian et al. [15] proposed an XML publish/subscribe system using a relational approach. Publish/Subscribe paradigm provides many-many communication. Candidates in a publish/subscribe-based communication system can act as a publisher or a subscriber of information. Publishers produce information sometimes called notifications, which is consumed by candidates which act as subscribers. A subscriber or consumer expresses their interests with the help of subscriptions for particular notifications. Relational database is used for matching purposes and evaluation of subscriptions is done using relational operations.

J. Shanmugasundaram et al. [19] showed ability of relational database to handle XML data type and proposed a system that performs various operations on XML data such as transformation of semi-structured queries into relational operators and SQL queries which helps to manage data efficiently. Relational database is used to process queries on XML documents.

Relational Database can be used to support many advanced operations.

C. Mayfield et al. [5] proposed a framework based on statistical analysis for data cleaning using database approach. While maintaining data it is the responsibility of the database to retain the consistency of the data. It should be as it is when it was stored into the database. Often there is case when some data changes or missed. A framework for deducing lost information and rectifying of information is proposed. A method is proposed to recover from the missing data in order to correct it based on statistical methods which execute repeatedly using inference algorithms which are implemented using relational databases based on regression and Bayesian networks.

An index structure is proposed for mining patterns for graphs which are too big to fit into main memory [13]. An algorithm is proposed ADI-Mine, which is used to build the index structure and contribute to scalability of graph mining.

S. Trißl et al. [11] proposed an index structure which can answer the reachability queries in almost constant time. Index structure is combination of SQL and graph operations providing advantage of integration since it is easy to combine it.

S. Srihari et al. [6], S. Chakravarthy et al. [10] proposed a framework for mining of huge graphs and sub graphs by using relational database system. Often large graphs are not efficient to store in main memory and need to be stored on disk and then fetched back to main memory when needed. Graphs are compressed to store and manipulate graph data in main memory. A SQL based framework is proposed to mine large graphs and processing on graphs which is mainly applicable to transactional datasets due to gap between relational context and graphs. It Combines graph applications with relational database for storing large graphs and uses SQL queries for querying and accessing of data.

Ruiwen Chen [3] proposed strategies of relational database to manage massive graphs with the help of framework which combines horizontal partitioning and graph partitioning with clustering algorithms. SQL query overhead is considered by making use of a new access method, Graph scan method by minimizing
access operations. [12] Showed the analysis of various social networking sites and how these sites handle this data.

Miao Qiao et al. [2] proposed a local landmark indexing approach which reduces indexing overhead and achieves optimization using graph compression and online search methods. Graph embedding techniques are used. Path with minimum cost is found by making use of triangle inequality property and relational operators. Considering the challenge of massive graphs to store in main memory, indexes are stored in relational database and queries to be performed are specified using relational operators. Kriegel et al. [9] proposed a hierarchical reference node embedding approach that achieves better scalability by arranging reference nodes in multiple levels.

III. FRAMEWORK TO FIND SHORTEST PATH IN RELATIONAL CONTEXT

Jun Gao et al. [1] proposed a relational method to find shortest path using three relational operators specified using relational algebra. A relational database based framework is proposed with the focus on performance of database system using advanced SQL features such as windows function and scalability is achieved using bidirectional algorithm. It follows selection-expansion-merge pattern for evaluating search queries. An algorithm is proposed which consists of three operations which finds shortest path between nodes using three operators. Search proceeds by selecting a node with minimum cost and expanding it to obtain newly expanded nodes which are then merged with visited nodes which leads to shortest path. RDB based framework includes following three steps:

1) Select Frontier node: Initially source is selected as frontier node. Then after expanding node with minimum cost it is taken as a frontier node.
2) Expand Frontier node: A node with minimum cost is expanded to find newly expanded nodes.
3) Merging: Visited nodes and expanded nodes are integrated which leads to shortest path.

Algorithm for shortest path estimation:

1. Initialize table with nodes
2. Locate ID for next frontier node
3. Expand Paths with frontier node
4. If dead end then break
5. Finalize the node
6. If result exists then break
7. Find shortest path
8. Return path

CONCLUSION

In this paper, we presented survey on graph operations and search queries in relational context. Shortest path can be computed using various existing methods but due to large data memory problems should continuously be addressed. Relational based approaches based systems can be used to bridge the gap graph operation and relational operations that can contribute to performance and scalability of the system.

REFERENCES


