ESTIMATION OF SHORTEST PATH FOR LARGE GRAPH USING RELATIONAL TECHNIQUE

Piyush Kulkarni,Kapil Vyas

PG Student,Computer Science And Engineering,BM College of Technology,Indor(MP),India

Assistant Professor, Computer Science And Engineering ,BM College of Technology,Indore(MP),India

ABSTRACT

This paper focus on computing shortest path as finding distance between two vertices in graph or tree. Graph data can use in many domains like social network and in knowledge graph. This graph search includes sub-graph. The problem of finding shortest path between two nodes can be solved using the salesman traveling path, minimal spanning tree, and the likewise. Problem occurred with the graph based searching when graph is too big to fit in memory and for that it uses the external memory. Disk-based method has some limitations when graph exceeds its size. In this paper, we are analyzing the shortest path for efficient relational approaches to graph search queries. For this, we use three relational operator based on which we introduce framework that bridge gap between graph operation and relational operator. We show new feature of SQL such as merge statement and windows function to improve performance of FEM framework. To avoid extra indexing overhead and improve scalability and performance, we propose an edge weight aware graph partitioned schema and design bi-directional restrictive BFS (breadth-first-search). The final experimental result illustrate our relational approach with optimization strategies can achieve high performance and scalability.

Keyword: *Graph,Shortest path, Relational database,FEM framework.*

1. INTRODUCTION

Today's world growth of graph increases rapidly, graph search faces more challenges. Graph search is more common in graph applications. Graph search purse for specific purpose such as shortest path between two nodes, minimum spanning tree etc. As size of graph increases, it does not fit into main memory so existing approaches to graph search must be reexamined. On external disk memory, I/O is the key factor for graph operations. As graph size increases, existing disk based methods provide limited support for graph based quires. Neo4J is one who can store large graph in database and provide operation such as shortest path discover to end users and graph traversal. The performance of graph based systems should be continuous improved, as graph database systems have to implements complex component including query evaluation, query optimization, storage etc. MapReduce framework and its open source implementation Hadoop can process large graph stored on distributed file system.

Relation database (RDB) provides support for graph search. RDB plays a key role in information systems. RDB and graph database management have many same functionality such as storage, optimization etc. RDB also managing complex database like XML data. RDB can support graph queries such as BFS and reachability query .The extension of RDB to graph search queries is useful when both graph and relational operation are needed.

It requires a substantial effort to support graph search query in RDB context. First, queries of graph search are various forms. It is not possible to implement each query. We need to find mechanism for evaluation of graph search in relational context. Second, there is symbolic mismatch between relational operations and graph operation

which effects on graph search. Graph operation follows node-at-time fashion whereas relational operations follow set-at-time fashion.

These paper focuses on shortest path discover for two reasons. First, search shortest path that provide key role in many application like reveal relationship between two individual in social network. Second it represent query which has similar evaluation pattern like other search queries.

2. LITERATURE SURVEY

B. Bahmani, K. Chakrabarti,et[1] Proposed a fast MapReduce algorithm. The basic idea behind this is single walk of length starting at each node in the graph G. The MapReduce algorithm is more efficient than other algorithm in MapReduce setting. The efficiency of MapReduce algorithm depends on many factors such as computation of job scheduler and data distribution. PPR approximation is implements using MapReduce algorithm which is outperforms of state-of-the-art algorithm. This algorithm uses less machine time and clock time.

Silke Trißl, Ulf Leser[2] proposed GRRIP index structure. This structure used for reachability queries on directed graph. This structure used to index graph with five million and more node as GRRIP requires only linear and time space. Using GRIPP, it is possible to answer reachability queries on any type of graph. GRRIP structure is based on SQL so it can be easily integrated in any existing graph application. GRRIP is fastest method for indexing large biological network.

B. Zou, X. Ma, B. Kemme,et[4] Proposed relational database as secondary storage to avoid limitation of handle limited amount of data. They proposed approach which provides a database interface for several algorithms. They use learning software package weka and added relational storage manager as back-tier to system. They referred wekaDB which allow any new algorithm that are added to weka package to be able to work immediately on data store in database without any modification.

M. Potamias, F. Bonchi,et[5] Discussed about landmark-based method for point to point distance estimation in large networks. This method selecting subset node as landmark and computing distance of each node in graph from those landmark offline. They proved the problem of optimal landmark selection is NP hard. They describe several techniques for landmark selection such as standard random and state-of-the-art technique.

H. Pirahesh ,F. Tian,et[6] design and implementing an XML publish/subscribe system to built an efficient, scalable engine t match published XML messages with millions of Xpath expressions using a relational database.

D. Wagner and T. Willhalm[7] Suggested various techniques to speed up the DIJKSTRA'S algorithm. These techniques return shortest path and run faster. Authors combine various speeds up technique such as bidirectional search, goal-directed search.

J.Dean and S.Ghemawat[8] Proposed MapReduce programming model. This model is easy to use for programmer without experience with distributed and parallel system. The number of optimization in system is targeted at reducing amount of data sent across the network. The redundant execution can be used to reduce impact of slow machine, and to handle machine failure and data loss.

E. Dijkstra[9] Focuses on two problems in connection with graphs and gives its solutions. Author considers nodes whose some or all pairs are connected by branch and length of each branch is given. Author gives solution of problem like construct tree of minimum total length between n nodes and find the path of minimum total length between two given nodes.

E. Cohen, E. Halperin,et[10] Introduced simple and natural distance reachability labeling schemes for undirected and directed graph. Labelings are derived from 2-hop covers of set of paths in graphs. Authors give an efficient algorithm for constructing a 2-hop cover whose size is larger than smallest 2-hop cover. They show effectiveness of their schemes by experimental analysis using real network.

D.Hutchinson,A.Maheshwari,et[11] Proposed an external memory data structure of shortest path queries. Authors store rooted tree in external memory for bottom u path can be traversed I/O efficiently. They discover various I/O efficient algorithms which can use for triangulating planner graph and computing separators of graph.

C.Wang, W.Wang,et[12] Developed index structure ADI (adjacency index). This Structure support various mining graph pattern over large database that cannot be held into main memory. The index structure can be easily adapted in various graph patterns mining algorithm. The index structure is faster than gSpan when both can run in main memory.

Harrelson, A Goldberg and [13] Proposed a new lower bounding technique based on landmark. They also give several landmark selection techniques. They uses A* algorithm in combination with new graph. They developed bidirectional variant of A* search and find variant of new algorithm to find most efficient. The algorithm compute shortest path and work on any directed graph.

C.Aggarwal, Y. Xie,et [14] Presented connectivity index for massive disk resident graphs. They create compress version of underlying graph and use it as index for processing query. This approach improves performance of system and provides high accuracy in terms of quality. Now they are working on graph stream approach.

C. Mayfield, J. Neville,et[15] Presented framework for missing information and correct them automatically. Their approach based on relational dependency network and include inference algorithm which is easily implemented in standard DBMS using SQL. The system uses shrinkage technique for cleaning task. The system is scalable and requires a minimum amount of domain knowledge.

R. Prims [16] Represents various algorithm for finding shortest path in graph such as, traveling salesman problem, minimum spanning tree. The aim of system is to maximize symmetric function of longest spanning sub-tree of connected graph.

Paper	Algorithm	Remark
[1]	MapReduce Algorithm	The basic idea behind this algorithm is single walk of length starting at each node in graph.
[2]	GRRIP index structure	This structure used for reach ability queries on directed graph .It is SQL based structure so easily integrated with existing graph application.
[3]	SQL based approach for querying and mining large graph	This approach own simple lightweight framework to work with graph application.
[4]	Data mining and machine learning Algorithm	The system uses relational database as secondary storage to avoid limitation of handle limited amount of data.
[5]	Shortest path Algorithm	The system proposed landmark based method for point to point distance estimation in network.
[6]	Content based Algorithm	Algorithm present XML implementation to built an efficient and scalable engine.
[7]	Dijkstra's algorithm	Algorithm return shortest path and run faster.
[8]	MapReduce algorithm	They introduced MapReduce programming model for distributed and parallel system.

[9]	Dijkstra's algorithm	The algorithm used for find minimum length between two given node.
[10]	Preprocessing algorithm	The system uses simple and distance reachability schemes for directed and undirected graph.
[11]	I/O efficient algorithm	The system proposes external memory as data structure for shortest path queries.
[12]	gSpan algorithm	The system proposed ADI index structure.
[13]	ALT- algorithm	The system proposed lower bound technique based on landmark. They uses A* algorithm.
[14]	GConnect algorithm	The system represent compress version of underlying and use it as index for query processing.
[15]	Noval approximation framework	Provide framework for search missing information and correct them automatically.
[16]	Prim's algorithm	This algorithm used for finding shortest path in graph.

3. CONCLUSIONS

This paper proposes FEM framework with three new operators. It also introduce new SQL feature such as merge function and windows function to improve performance of FEM framework. . If we distribute the database over multiple systems, efficiency of system can be improved. The bidirectional restrictive search and graph table is partitioned which improves the scalability and performance of FEM Framework.

4. REFERENCES

[1] B. Bahmani, K. Chakrabarti, and D. Xin, "Fast Personalized Pagerank on Mapreduce," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '11), pp. 973-984, 2011.

[2] S. Trißl and U. Leser, "Fast and Practical Indexing and Querying of Very Large Graphs," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'07), pp. 845-856, 2007.

[3] S. Srihari, S. Chandrashekar, and S. Parthasarathy, "A Framework for SQL-Based Mining of Large Graphs on Relational Databases," Proc. 14th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining—Volume Part II (PAKDD'10), pp. 160-167, 2010.

[4] B. Zou, X. Ma, B. Kemme, G. Newton, and D. Precup, "Data Mining Using Relational Database Management Systems," Proc. 10th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining ('06), pp. 657-667, 2006.

[5] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis, "Fast Shortest Path Distance Estimation in Large Networks," Proc. Int'l Conf. Information and Knowledge Management (CIKM'09), pp. 453-470, 2009.

[6] F. Tian, B. Reinwald, H. Pirahesh, T. Mayr, and J. Myllymaki, "Implementing a Scalable XML Publish/Subscribe System Using a Relational Database System," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '04), pp. 479-490, 2004.

[7] D. Wagner and T. Willhalm, "Speed-Up Techniques for Shortest- Path Computations," Proc. 24th Ann. Conf. Theoretical Aspects of Computer Science (STACS '07), pp. 23-36, 2007.

[8] J. Dean and S. Ghemawat, "Mapreduce: Simplified Data Processing on Large Clusters," Proc. Sixth Symp. Operating System Design and Implementation (OSDI'04), pp. 137-150, 2004.

[9] E. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, vol. 1, pp. 269-271, 1959.

[10] E. Cohen, E. Halperin, H. Kaplan, and U. Zwick, "Reachability and Distance Queries via 2-Hop Labels," Proc. 13th Ann. ACMSIAM Symp. Discrete Algorithms (SODA '02), pp. 937-946, 2002.

[11] D. Hutchinson, A. Maheshwari, and N. Zeh, "An External Memory Data Structure for Shortest Path Queries," Discrete Applied Math., vol. 126, pp. 55-82, no. 1, 2003.

[12] C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi, "Scalable Mining of Large Disk-Based Graph Databases," Proc. 10th ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '04), pp. 316-325, 2004.

[13] Goldberg and C. Harrelson, "Computing the Shortest Path: Search Meets Graph Theory," Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '05), pp. 156-165, 2005.

[14] C. Aggarwal, Y. Xie, and P. Yu, "GConnect: A Connectivity Index for Massive Disk-Resident Graphs," Proc. VLDB Endowment, vol. 2, no. 1, pp. 862-873, 2009.

[15] C. Mayfield, J. Neville, and S. Prabhakar, "ERACER: a Database Approach for Statistical Inference and Data Cleaning," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIMGMOD '10), pp. 75- 86, 2010.

[16] R. Prim, "Shortest Connection Networks and Some Generalizations," Bell System Technical J., vol. 36, pp. 1389-1401, 1957.