



**International Journal of Intellectual Advancements
and Research in Engineering Computations**

**EFFICIENT TRANSITIVE REGION APPROACH FOR SHORTEST PATH
DISCOVERY IN RDBMS USING MINIMUM SPANNING TREE**

¹M.Helen Freda, ²S.Prema M.E.,

ABSTRACT

This paper aims to find the shortest path with the help of efficient ETR approach to graph search queries. We have proposed an Efficient Transitive Region (ETR) a performance-based initiative that utilizes the SP transitive regions to construct the shortest path and also to improve it. Here shortest path construction systems as dominated the path overlapping. ETRR constructs an auxiliary shortest path step by step, instead of considering single step at a time using Shortest Path trees during traversals.

KEYWORDS: ETR approach, KWS interface, shortest path Shortest Path trees.

INTRODUCTION

With the very fast growth of graphs, it faces many challenges. Usually graph search means it finds the sub-graph that meets the specified conditions like minimum spanning tree, travelling salesman, web graph etc. Due to this huge growth it finds it difficult to fit into main memory, and also the existing in-memory methods cannot be used due to I/O overhead. This restriction limits them in implementing for larger graphs. Neo4j can be used because it has the capability to store large graphs.

The graph database systems has to be mature and stable enough to support complex operations, components, including storage, indexing, evaluation, and optimization. For example, in planar graphs it should have the capability to generate shortest path index for external memory. MapReduce framework and its open source implementation Hadoop usually has the capability to support large graphs stored in a distribute file system, but here the main concern is the low latency of the users during the design of hadoop.

Relational databases are very mature and stable enough to form a solid foundation to support general search queries. It has the capability of supporting large, complex datatypes and also has the ability to support functions such as indexing,

data storage, buffering etc. They integrate graph and relational operations together, but they cannot be implemented independently to support the various forms of graph search queries. There is also semantic variation in its functionality which affects the performance of graph search. It can process only a set at a particular time such as projection, selection, and join etc. Usually shortest path discovery is carried out to find the relationship between two individuals, for finding path with least distance and also to be used for other graph search queries.

We have proposed an Efficient Transitive Region (ETR) a performance-based initiative that utilizes the SP transitive regions to construct the shortest path and also to improve it. Here shortest path construction systems as dominated the path overlapping. ETRR constructs an auxiliary shortest path step by step, instead of considering single step at a time using Shortest Path trees during traversals.

ETRR when used along with graph clustering technique, which usually requires graph analyses thereby grouping similar vertices together. It introduces an optimization strategy based on sub-graph traversal to improve the performance of the clustering process. It also helps in achieving scalability without incurring extra overhead and index construction.

Author for Correspondence:

¹Department of CSE, Mahendra Institute of Technology, Namakkal, Tamilnadu, India.

Email: helen.freda@gmail.com.

²Department of CSE, Mahendra Institute of Technology, Namakkal, Tamilnadu, India.

Email: sprema074@gmail.com.

SHORTEST PATH COMPUTATION USING ETR APPROACH

KEYWORD INDEXES

In this module ETR addresses keyword index for both generation of structured queries and generation of answer tuples by using cell-level granularity term frequency information and column-level granularity term frequency information. Transitive region Granularity and Lab Cell Granularity are the two relational tables used to store term frequency information at column and cell granularity level.

ETR Building main memory hash map of column-level granularity term frequency.

- For keyword query and ranking, generate a candidate structured queries.

- To obtain a structured query, generate a corresponding SQL query and calculate relevance scores for each of the answer tuple.

GENERATION OF ANSWER TUPLES

In this module providing full-text index requires huge storage space, every string attribute in the domain database may not have full-text indexes. But for providing KWS interface, without any of the full-text indexes on published attributes results in sequential scan of the corresponding relational tables. So ETR maintains separate cell-level granularity term frequency information, without changing domain database, for those published attributes which do not have full-text index and avoids costly sequential scan at runtime.

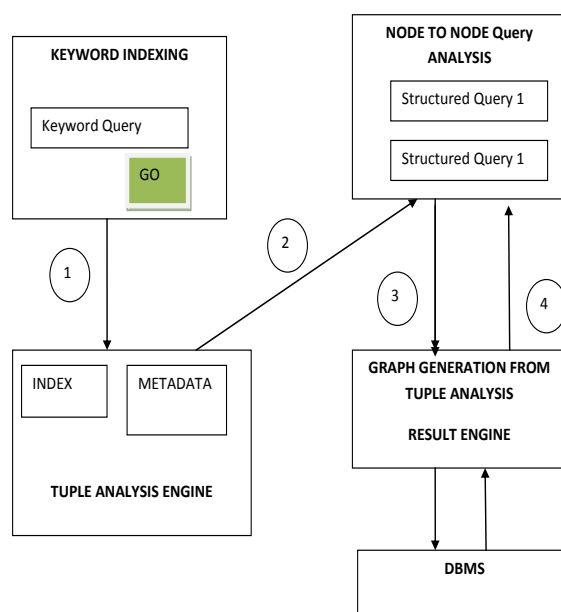


Fig1: Architecture Diagram

METADATA GRAPH GENERATION

In this module Metadata in the Graph Generalizing this idea, it is not difficult to envisage other useful, perhaps analytical, modules that would benefit from writing extra information into the graph. There is no reason such "metadata" should be limited to node properties; they could take the form of relationship properties, but also entire sub graphs. The framework, for the moment acting only as a library of useful code, provides a set of interfaces and abstract base classes to support such a development.

RELATED WORK

GRAPH SEARCH

Graph queries are very much essential for carrying out graph operations, but as the search space becomes large they cannot be solved based on these graph queries alone. So methods like Dijkstra, Prim and travelling salesman algorithm, 2-Hop can be used, as they compute the shortest path online and offline they play a very important role in boosting the performance in run time. But

the main drawback is that they cannot be used for graphs that cannot be fitted into main memory.

EXTERNAL GRAPH OPERATIONS

The external graph operations such as planar graphs, MapReduce Framework, and its open source implementation Hadoop as the capability to support large graphs and also provide high scalability. But the main drawback of Mapreduce Framework is that it does not have the capability to support online queries and also faces difficulty in performing updates on graphs. So it is very difficult to implement external operations for graph search queries.

RDB BASED GRAPH OPERATIONS

RDB is a stable foundation to support complex data, statistical data analysis, sophisticated applications, and also in this application can be converted to sequence of SQL. This conversion is very important because it is optimized to fewer SQL's. Transitive closure, aggregation and selection can be used to find shortest path but each one uses its own method to find shortest path. Here aggregation follows recursion pattern to find the shortest path. In this the shortest path is found with new features of SQL, restrictive BFS and the table partitioning schemas. The reachability query is examined by a stored procedure, SQL based graph data mining approaches etc. And also generic graph search framework is designed in RDB, which improves the efficiency of the framework with new features of SQL. The relational shortest path discovery is optimized by reducing the search space, friendly evaluation of RDB and table partitioning schema.

CONCLUSION

This paper aims to find the shortest path with the help of efficient ETR approach to graph search queries. We have proposed an Efficient Transitive Region (ETR) a performance-based initiative that utilizes the SP transitive regions to construct the shortest path and also to improve it. Here shortest path construction systems as dominated the path overlapping. ETRR constructs an auxiliary shortest path step by step, instead of considering single step at a time using Shortest Path trees during traversals. A performance analysis of ETR carried out on real and artificial graphs reveals that due to the large number of shortest path with least distance the execution time

has increased to 80 %, and also the performance is improved when compared to that of the FEM framework approach. The ETR with multiple SP computations reduces the latency, thereby providing high efficiency and scalability when compared with the existing methods. Due to space limitations, the extension of the ETR framework to other complex graph operations will be our further work.

ACKNOWLEDGMENT

I would like to thank the Head of the Department, guide of my institution and my family members and friends who rendered all possible encouragement and support during the course of this project.

REFERENCES

- [1]. J. Gao, R. Jin, J. Zhou, J. Xu, X. Jiang, and T. Wang, "Relational Approach for Shortest Path Discovery over Large Graphs," Proc. VLDB Endowment, vol. 5, no. 4, pp. 358-369, 2011.
- [2]. 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.
- [3]. 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 (SIGMOD '10), pp. 75-86, 2010.
- [4]. J. Cheng, Y. Ke, A.W. Fu, J.X. Yu, and L. Zhu, "Finding Maximal Cliques in Massive Networks by H*-graph," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '10), pp. 447-458, 2010.
- [5]. 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.
- [6]. 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.
- [7]. A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee,

“Measurement and Analysis of Online Social Networks,” Proc. Seventh ACM SIGCOMM Conf. Internet Measurement (IMC '07), Oct. 2007.

- [8]. A. 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.
- [9]. 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.
- [10]. 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.