International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015
ISSN 2091-2730

Magnified Coherency Technique for Effective Relational Keyword Search

SURYA SUGUNAN

Department of Computer Science and Engineering

Sarabhai Institute of Science and Technology

Vellanad,Trivandrum,India

Sugunansurya90@gmail.com

Mob No-8281545272

Abstract-In the past decade, extending the keyword search paradigm to relational data has been an active area of research within the database and information retrieval community. Many approaches have been proposed but there remains severe lack of standardization for system evaluations. This paper presents the most extensive empirical performance evaluation of relational keyword using an enhanced coherency technique in a peer to peer system. In this paper consider a peer to peer system having same databases for relational keyword searching and evaluations. From one remote system we can directly access all the other relational databases that is connected to a network and evaluate the efficiency of new proposing searching algorithm with traditional searching method. The results indicate that traditional approach have little time consuming whenever the number of nodes increases. In summary my work confirms the unacceptable performance of traditional approach by using an enhanced coherency technique.

Keywords-Relational data, performance evaluation, tuples, sampling, DISCVOVER, coherency technique, random walk.

1. INTRODUCTION

Keyword search is the most popular information discovery method. When a set of keyword is provided by the user, the search engine results all the documents that are related with these keywords. Unfortunately, keyword search techniques used for locating information from collections of web documents cannot be used on data stored in databases. In relation databases, information needed to answer a keyword query is split across the tuples in tables, due to normalization. This paper describes the keyword search in relational databases in a peer to peer system using DISCOVER algorithm. It explores the most extensive empirical performance evaluation of relational keyword search compared with the traditional approach. Typically the user of a relational database needs to know the schema of the database SQL or QBE-like interface, and the roles of the different entities and terms used in the query. The user of the DISCOVER does not need a vast knowledge of any of the above. It enables a straight forward keyword search interface to the database. DISCOVER does not require from the user to know the relations and the attributes where the keywords are found.

This paper present a thorough empirical performance evaluation of relational keyword search system dynamically based on the benchmark and introducing the DISCOVER algorithm, which is more powerful and efficient than the existing keyword search technique in relational databases.

Overview of Relational Keyword Search

The implicit assumption of keyword search is, the search terms are related complicates the search process because typically there are many possible relationships between two search terms. This realization leads to little tension between the compactness and coverage of search results. This paper proves that DISCOVER finds without redundancy all relevant candidate networks by exploiting the structure of the schema.DISOVER operates on relational databases and facilitates information discovery on them. A high level representation of the architecture DISCOVER used to find the joining networks shown in figure 1.

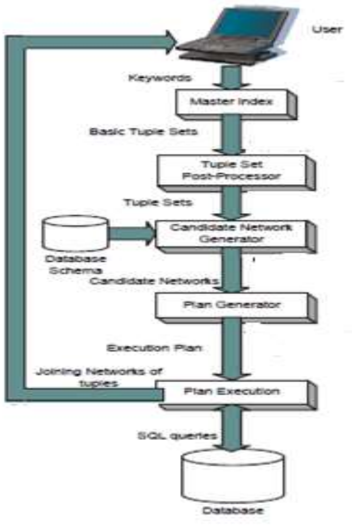

Figure 1.Architecture of discover algorithm.

First user gives a set of keywords k1,…..km to the system. These keywords are looked up in Master Index, which returns the tuple sets $R_i^{k}{}_{1}$,….$R_i^{k}{}_{m}$ for each relation R_i.Every tuple of $R_i^{k}{}_{j}$ contains keyword K_j as part of an attribute value. Then DISCOVER calculates all candidate networks. The set of candidate networks is guaranteed to produce all the minimal joining networks. Then DISCOVER evaluates the candidate networks. The candidate networks share join to expressions. So this causes an opportunity to build a set of intermediate results and use these in the calculation of multiple candidate networks. The Plan Generator generates an execution plan that calculates the intermediate results for evaluating the candidate networks. Finally an SQL statements are passed to the DBMS.Then DBMS returns the joining networks of tuples that are the solution to the query problem.

Contribution of this paper.

- Propose a cost model. The plan generator module uses intermediate results to minimize the total cost of the evaluation of all candidate networks.
- Prove that candidate network generation algorithm creates a complete(the set of candidate networks produces all minimal joining networks of tuples up to a given size T) and non-redundant(if any candidate network of the set is excluded then there are DB instances where there are minimal joining networks of tuples that are not discovered)sets of candidate networks.

2. Related Works

A framework for keyword search on databases when the schema is not known to the user is presented in [4] [5].An extension of SQL called Reflective SQL (RSQL) is treats data and queries uniformly. The main limitation of this work is that all keywords must be contained in the same tuples. BLINKS [3] an indexing and query processing scheme for ranked keyword search over node-labeled directed graphs. It uses the backward search strategy of BANKS [6] but based on cost-based expansion .BLINKS needs to use separate cursors not just for each keyword cluster but also for each block that it has to traverse. So overhead of maintaining these cursors adversely affect the overall performance. BLINKS lack approximation and run-time guarantees because of the use of indexes and a different metric. STAR [2] runs simple breadth first search iterators from each terminal instead of running single source shortest path iterators from each node of v in BANKS.STAR has to maintain only two iterators per improvement step so we have to avoid performance degradation. It uses fairly tight upper bounds on the lengths of the paths and prunes the possible paths that can be included in the result tree. It achieves o(log n) approximation for the optimal Steiner tree problem. The drawback of this approach is that a graph of the tuple must be created and maintained for the database.

In contrast DISCOVER is tuned to keyword search on relational databases and uses the properties of the schema of the database. Its main algorithm works on the schema graph and it does not need to keep any extra data representations. It produces the minimum number of SQL queries needed to answer to the keyword query. DISCOVER operates directly on the databases, so it does not have a main memory space limitation.

The use of common sub expressions by the plan generator is a form of multi-query optimization [7][8][9].The candidate networks in DISCOVER have special properties that allows to develop a more better straight forward and efficient algorithm. The first property is that the candidate networks have small relations as leaves. Second, the candidate networks are not random queries. The existing search techniques are adhoc with little standardization. The previous work [11] compares relational keyword search techniques but does not consider run time performance. So this work aims to evaluate the runtime performance of relational keyword search technique with DISCOVER method dynamically in a peer to peer system.

3. Frameworks

3.1 Data Model and Keyword Queries

Consider a database that has n relations $R_{1,\dots}R_{n.}$Each relation R_i has m_i attributes $a_1^i\dots a_{mi}^i$.The schema graph G is a directed graph that captures the primary key to foreign key relationships in the DB schema. So an edge $R_i \rightarrow R_j$identifies the corresponding primary and foreign key attributes. Assume that no sets of attributes of any relation in schema are both a primary key and foreign key for two other relations in database schema.

Definition 1 (Joining network of tuples in a relational schema)-A joining network of tuples j is a tree of tuples where for each pair of adjacent tuples $t_{i,}t_j \in j$ where$t_i \in R_i$ and $t_j \in R_{j,}$there is an edge $(R_{i,}R_j)$ in G_u and $(t^i \bowtie t^j) \in (R_i \bowtie R_j)$.

Definition 2 (Keyword Query)-A keyword is a set of keywords $k_{1\dots}k_{m.}$The result of the query is the set of all possible joining networks of tuples in the schema that are both minimal and total.

Definition 3 (Joining network of tuple sets in a relational schema)-A joining network of tuple sets J is a tree of tuple sets $R_i^K R_j^M$ in J there is an edge $(R_{i,}R_j)$ in G_u.

Definition 4 (Candidate network)-A candidate network C is ajoining network of tuple sets ,such that has a MTJNT(minimal total joining networks of tuples) M ∈ C and no tuple t∈ M that maps to a free tuple set F∈ C contains any key words.

Definition 5 (Execution Plan)- Consider a set of candidate networks $C_{1\dots}C_r$ an execution plan is a list of assignments $A_1\dots A_S$of the form $H_i \leftarrow B_i, \bowtie \dots B_i t$.

3.2 Architecture of DISCOVER

In this section, formally define the architecture of DISCOVER algorithm.

The Master Index inputs a set of keywords $k_{1\dots}k_{m.}$and outputs a sets of basic tuple sets denoted as $R_i^{-k}{}_j$ for i=1…..n and j=1….m.It consist of all tuples of relation $R_i$that contain the keyword k_j.The master index is implemented using the Oracle8i interMediaText8.1.5 extension. It builds full text indices on single attributes of relations. Then it inspects the index of each attribute and combines the results. Then the tuple set post-processor takes the basic tuple sets and produces tuple sets R_i^k for all subsets k of { $k_{1\dots}k_m$}.The non-empty tuple sets passed to the candidate network generator. Then the set of candidate networks transferred to the plan generator, which optimizes the evaluation of networks. Finally the execution plan is going to the Plan Execution module, which translates the assignments of the plan to SQL.The assignments that build intermediate results are translated to "CREATE TABLE" statements and the candidate network evaluation assignments to "SELECT-FROM-WHERE" statements is the result of the keyword search and the result is returned to the user.

4. Candidate Network Generation

Candidate network generator outputs a complete and non-redundant set of candidate networks. Ensures that the joining networks of tuples are total and minimal.

4.1Candidate networks generation algorithm

```
Algorithm Candidate Networks Generator
Input: tuple set graph GT S , T , k1 , . . . , km
Output: set of all candidate networks in the relational database with size up to T
{
  Q: queue of  activejoining networks of all possible tuple sets in a relation
  Pick a keyword kt∈ {k1 , . . . , km }
for each tuple set $R_i^K$ where i = 1. . . n and kt∈ K do
   Add joining networks of tuple sets $R_i^K$ to Q
while Q not empty do {
   Get head C from Q
if candidate network C satisfies the pruning condition of the Discover  then ignore C
else if C satisfies the acceptance conditions of the DISCOVER then output C
   /*There is no reason found to extend accepted joining networks of tuple sets in a relation*/
else
for each tuple set $R_i^K$ adjacent in GT S (ignoring edge direction) to a node of C
if (K = {} OR$\nexists R_J^M$ ∈(C∪ $R_I^K$,M≠{} ∧ keywords(C ∪$R_i^K$ ) = keywords((C ∪$R_i^K$ ) −$R_j^M$ ))
   /*Expansion rule*/
and (size of C < T ) then {
if$R_{ii}^K$ is adjacent to $R_J^M$ C = $R_J^M$ [. . .] then C ← $R_i^K$ [$R_j^M$ [. . .]]
          Put C in Q
       }
else ignore $R_I^K$
  } }
```

5 Evaluation of candidate networks

.

Cost Model- Assign a cost of 1 to each join. The actual cost model of DISCOVER exploits the fact that we can get the sizes of the non-free tuple sets from the master index. The cost of the execution plan is the sum of the costs of its joins.

5.1Greedy Algorithm

It produces a near optimal execution plan, with respect to the actual cost model.

Algorithm Select list of intermediate results generated by the Naïve method

Input:set S of candidate networks of size ≤T

Output: Produce a list L of intermediate results to build for the evaluation.

```
{

While not all candidate networks generated in S have been added to L do
{
Let Z be the set of all small join sub expressions of 1 join contained in at least one candidate network in S ;
Add the intermediate result m with the maximum value in Z to L;
Rewrite all candidate networks in S to use m with possible;
}
}
```

6 . Experiments

Consider a peer to peer system having the same SQL databases. Using the sampling mechanism to find out the active nodes in the network. Random walk is used to find out the visited and non-visited nodes based on the systems having SQL database or not. It is shown in figure 4 and 5. Sampling is one of the basic statistical tools used heavily in information integration and knowledge discovery fir approximating results from large datasets.

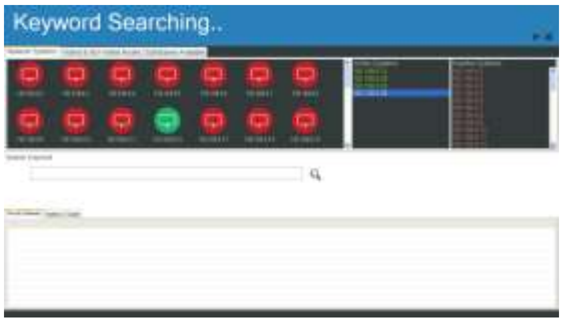

Figure 4 Active systems

Random walk based techniques have been developed to collect uniform samples from the distributed data. In a graph G random walk is a sequence of nodes visited where at each step the next destination node is selected with a fixed probability.

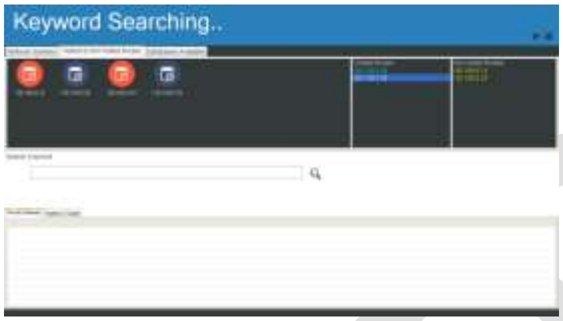

Figure 5 Visited nodes

In tuples clustering apply the fast clustering method to the classification of the database tuples. The similarity of sample is measured by distance. Then implement the DISCOVER algorithm to find out the relevant data from all the connected databases in a network from one remote system. Then analyse the results with the traditional approach and DISCOVER algorithm. Performance evaluation use two metrics to measure the run time performance. Execution time and response time. Execution time ,which is the time elapsed from issuing a query until an algorithm terminates because there are large number of potential results for each query, search techniques typically returns only the top-k results where k specifies retrieval depth. The response time is the time elapsed from issuing the query until I results have been retrieved. The graph of response times of both techniques (traditional and DISCOVER) is plotted in figure 6.

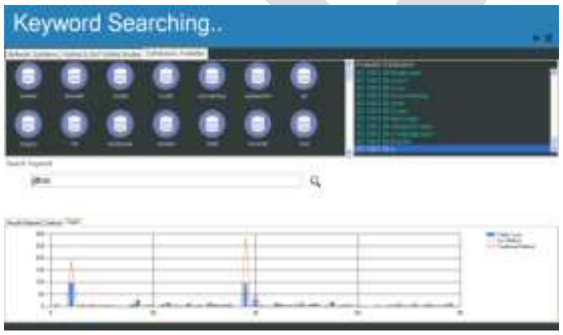

Figure6..Response time comparison.

Then the memory is measured for both the techniques. The schema based systems consumes very little memory. In contrast graph based approaches require considerably more memory to store their data graph.

Figure 7 Table counts.

Then calculate the no of counts of tables for searching the keyword in both methods. From the results we can conclude that the DISCOVER algorithm is little accurate than the traditional method.

ACKNOWLEDGEMENT

This work was supported in part by the Department of Computer Science & Engineering, SIST, Trivandrum. I would like to show my gratitude to Dr. C G Sukumaran Nair (HOD), Associate Professor, Sudha SK and Assistant Professor, Lizmol Stephen, for their valuable guidance.

7 .CONCLUSION AND FUTURE WORK

The importance of internet and internet users increasing day by day. Internet users increasingly demand keyword search interfaces for accessing information and it is natural to extend this paradigm to relational data. The lack of technology transfer with discrepancies among existing evaluations indicates a need for independent evaluation of search techniques dynamically in relational databases with minimum cost and maximum efficiency.

The DISCOVER algorithm covers the drawbacks of traditional method and it gives an opportunity to the user to extract the data easily without knowing the detailed knowledge of the query language.

REFERENCES:

[1] V. Hristidis and Y. Papakonstantinou, “DISCOVER: KeywordSearch in Relational Databases,” Proc. 28th Int’l Conf. Very Large Data Base (VLDB ’02), pp. 670-681, Aug. 2002.

[2] G. Kasneci, M. Ramanath, M. Sozio, F.M. Suchanek, and G.Weikum, “STAR: Steiner-Tree Approximation in Relationship Graphs,” Proc. Int’l Conf. Data Eng. (ICDE ’09), pp. 868-879, Mar.
2009.

[3]H. He, H. Wang, J. Yang, and P.S. Yu, “BLINKS: Ranked Keyword Searches on Graphs,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’07),pp. 305-316, June 2007.

[4] U. Masermann and G. Vossen. Schema IndependentDatabase Querying (on and off the Web).*Proc.Of IDEAS2000*, 2000.

[5] Ute Masermann and Gottfried Vossen. Design and Implementation of a Novel Approach to Keyword Searching in Relational Databases.*ADBISDASFAA Symposium*, 2000.

[6] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S.Sudarshan, “Keyword Searching and Browsing in Databases Using BANKS,” Proc. 18th Int’l Conf. Data Eng. (ICDE ’02),pp. 431-440, Feb. 2002.

[7]Timos K. Sellis. Multiple-query optimization.*TODS*, 13(1):23–52, 1988.

[8]Sheldon J. Finkelstein. Common subexpressionanalysis in database applications.*ACM SIGMOD*,
1982.[9]Prasan Roy, S. Seshadri, S. Sudarshan, and SiddheshBhobe. Efficient and extensible algorithms for multi query optimization. *SIGMOD Record 29*(20;249-260,2000.

[10]J. Coffman and A. C. Weaver, "A Framework for EvaluatingDatabase Keyword Search Strategies," in Proceedings of the 19th ACMInternational Conference on Information and Knowledge Management,ser. CIKM '10, October 2010, pp. 729–738. [Online].Available:http://doi.acm.org/10.1145/1871437.1871531

[11]A. Baid, I. Rae, J. Li, A. Doan, and J. Naughton, "Toward Scalable Keyword Search over Relational Data," Proceedings of the VLDB Endowment, vol. 3, no. 1, pp. 140–149, 2010
[12]Y. Chen, W. Wang, Z. Liu, and X. Lin, "Keyword Search on Structured and Semi-Structured Data," in Proceedings of the 35th SIGMOD International Conference on Management of Data, ser. SIGMOD '09,
June 2009, pp. 1005–1010