ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (128-131), Month: July - September 2015, Available at: www.researchpublish.com

Keyword Query Routing Using MHR Tree

Prachi Karale[1], Natikar S. B.[2]

[1]Student, Savitribai Phule Pune University,[2]Assis.Prof, Savitribai Phule Pune University,[1,2]VACOEA, Ahmednagar, India

Abstract: **Keyword queries are difficult to handle by any nontechnical user. For that purpose existing system used routing plans. KERG (Keyword Element Relationship Graph) is used in existing system and after that joins between two pair of keywords are used to get related resources. But as the number of keyword increases, number of joins in existing system increases. So here proposed system is MHR tree. Markle tree is created and after giving keyword query, keywords are searched using hash keys. Hash keys are same for sources and finally we give result along with these keys.**

Keywords: KERG, MHR, Routing Plan.

I. INTRODUCTION

THE web is not only a collection of textual documents but also a web of interlinked data sources (e.g., Linked Data). Linked data keywords are related to each other. If any user has given task to find out keywords within this linked data, then is difficult for the typical web users to exploit this web data by means of structured queries using languages like SQL or SPARQL. To end this, keyword search has proven to be intuitive. No structured queries, no knowledge of the query language, the schema or the underlying data are needed in keyword search.

In existing system, dataset is arranged in the form of sources, sets, elements and keywords (Keyword Element Relationship Graph)

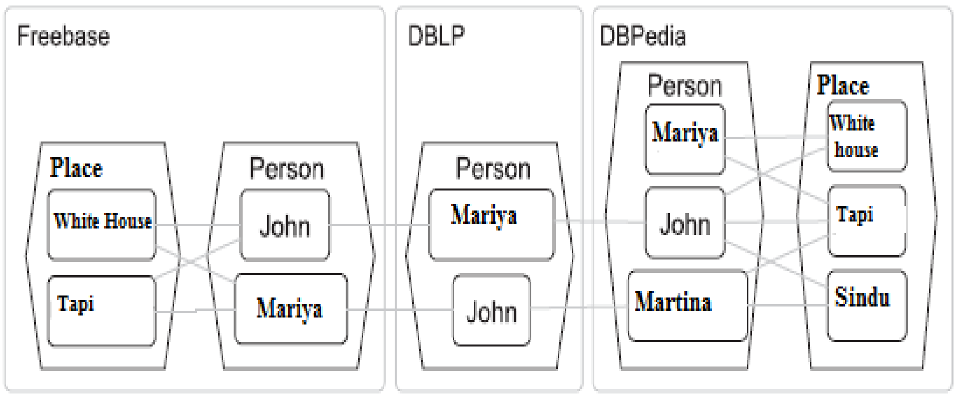

Fig.1 Set Level KERG

For obtaining KERG for given query understand the fig 1. It has three sources (Freebase, DBLP, DBPedia). Freebase contains two sets place and person, Place set consist of white house and tapi keywords. DBLP contains one set person. DBPedia have two sets person and place. Each keyword is related to another set keyword as shown in figure. From figure get relationship between different sources.(John, Person, Freebase) is Keyword element node. (John, person, DBPedia), (White house, Place, DBPedia) is keyword-element relationship, actually stands for the element-level connections.

In existing system, if keyword query is given as k={john, white, house}. Then we take two keywords pair such as {john, white}, {white, house},{john, house}. And after that take join of these pairs. And then routing plan derived from this is {Freebase, DBpedia}.

But as the number of keywords increases then number of joins increases, so time required is greater. So here proposed MHR tree (Markel Hash Tree). MHR tree is based on R-tree augmented with the min-wise signatures and linear hashing techniques. In case of MHR tree dataset is arranged in form of tree such that at the higher most level Sources are there, after that Set, then element and lastly keyword. If any keyword query is given then find out keyword location i.e.

Leaf node or root node. If at root node then find out which child entry gives maximum intersection with query and then compare with query keywords to get the keyword which is related to keyword query. And if child node then the above process which is related to child entry, is done over child node.

II. RELATED WORK

Previously keyword search is to find out result related to each and every keyword in keyword query i.e. it gives structured result [1], [2], [3], [7], [8]. But as they find out searching result for each and every keyword, space required for this is very large. Because this system is convenient for small number of sources, as number of sources increases then the potential result may increases exponentially with it. And most of the results are not required by users as this search does not gives exact information about keyword in keyword query, instead it gives each and every keyword related information in query.

So, M-KS [5] and G-KS system are used later for keyword search. M-KS captures relationships using a matrix. This is the case when all query keywords are pair wise related but there is no combined join sequence which connects all of them. Keyword relationship matrix is used here. G-KS [6] addresses this problem by considering more complex relationships between keywords using a keyword relationship graph (KRG).After comparison between M-KS and G-Ks, G-KS gives better result. But both of them give single source which contains each and every keyword of keyword query. This system do not think sources can be inter-related and required result is not related to single sources for that purpose existing system used routing plans. Routing plans uses joins for each and every two keyword pair. These join result gives related sources and using these related sources they can find out keyword search more easily as compared to search within each and every source. But as the number of keyword in keyword query increases then the number of joins increases to resolve this problem MHR tree is proposed.

III. MHR TREE

MHR Tree:

MHR-tree is efficiently answering approximate string match queries in large spatial databases. The MHR-tree is based on the R-tree augmented with the min-wise signature and the linear hashing technique. For a range query r, start from the root and check the MBR of each of its children, then recursively visit any node u whose minimum bounding rectangle (MBR) intersects or falls inside r. When a leaf node is reached, all the points that are inside r are returned. R*-trees achieve better performance in general than the original R-trees. This is general view of MHR tree.

MHR tree is created using whole dataset given for searching and arranged depending on their ranges. For searching it takes leaves keywords and then search within them (keywords). Here we are using hashing for searching the keywords. For each source passed same key and whenever we give any keyword query then flag values (number of times keywords find out) along with their key values get displayed.

Mathematical modeling:

Input: keywords query

Output: Search result

Process:

1) Tree Construction: R - Root of tree

L - Leaf nodes of tree

H - Hash for each node

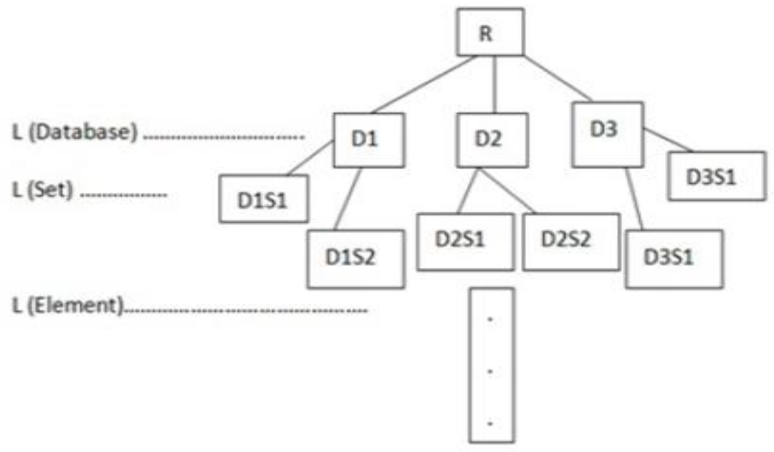

Fig.2 Block diagram of MHR tree

2) Tree Traversing: Tree traversing is done using hash key of each node. Each node has hash key and keyword will be search using hash keys which are identical to related set and element.

IV. RESULT

Fig. 3 shows time required for three keyword search in this case routing plans are best solutions. Fig. 4 gives time required for 7 keywords. Here we find out as number of keywords increases then hashing technique used in MHR tree gives best result as compared to existing system.

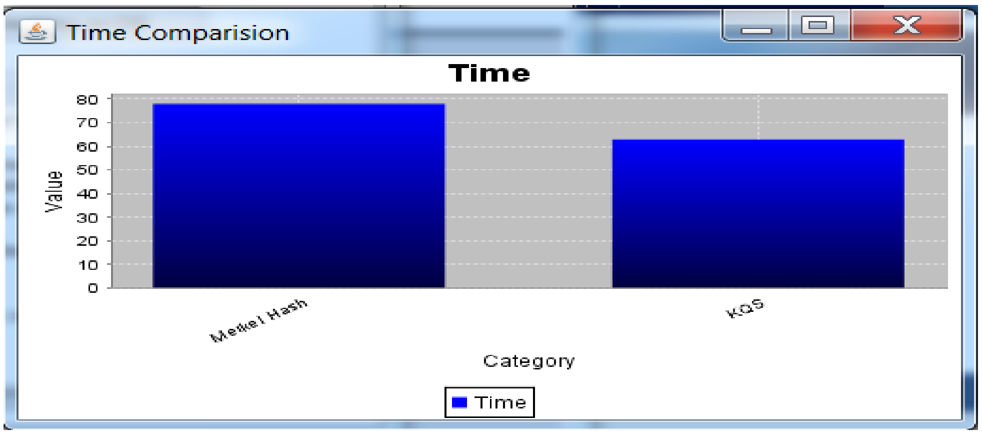

Fig. 3 Keyword search for 3 keywords

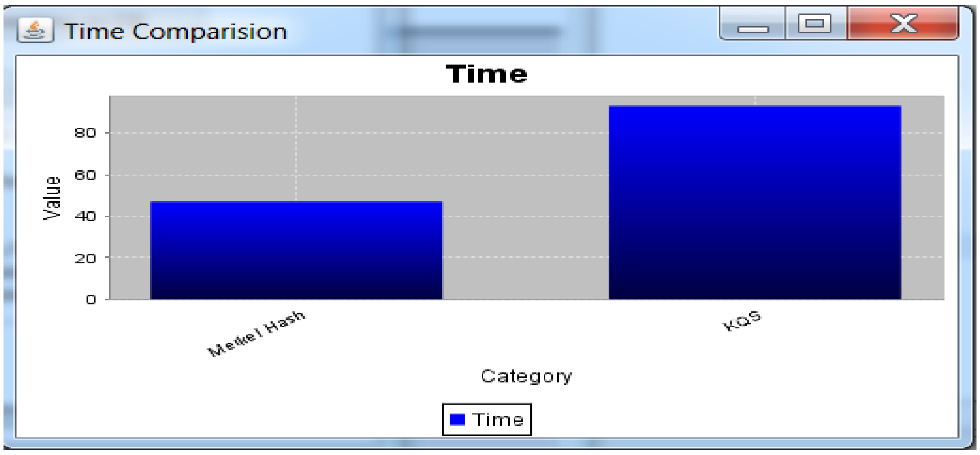

Fig. 4 Keyword search for 7 keywords

V. CONCLUSIONS

Existing system used routing plans for keyword routing. But as the number of keywords increases then number of joins increases. So, we proposed MHR tree .MHR tree uses hashing techniques and min-wise signatures. As MHR tree data is arranged in tree form before giving any query and after we give any query then we get hash code related to keywords as hashing techniques are used. Using these hash code we find out related result to given query. In tree it is easy to add branch or remove. So pruning is easy in MHR tree as compared to routing plans.

ACKNOWLEDGEMENTS

I thank to my project guide Prof. Natikar S. B. whose guidance and encouragement contributed greatly to the completion of this paper. I express my gratitude to ME Coordinator Prof. Kshirsagar M. and Head of Computer Department Prof. Mrs. Sarika Joshi of Vishwabharti Academys College of Engineering, Sarola Ahmednagar. I would like to thank to all faculty members Mr. Prabhudev, Mrs. Jagtap Madam of Computer Engineering Department and all my friends for their co-operation and supports. I would also like to thank our principal Dr. A. K. Kureshi who creates a healthy environment for all of us to learn in best possible way.

REFERENCES

[1] V. Hristidis, L. Gravano, and Y. Papakonstantinou, Efficient IR-Style Keyword Search over Relational Databases, Proc. 29th Intl Conf. Very Large Data Bases (VLDB), pp. 850-861, 2003.

[2] F. Liu, C.T. Yu, W. Meng, and A. Chowdhury, Effective Keyword Search in Relational Databases, Proc. ACM SIGMOD Conf., pp. 563-574, 2006.

[3] T. Tran, H. Wang, and P. Haase, Hermes: Data Web Search on a Pay-as-You-Go Integration Infrastructure, J. Web Semantics, vol. 7, no. 3, pp.189-203, 2009.

[4] Thanh Tran and Lei Zhang, Keyword Query Routing, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 2, February 2014.

[5] B. Yu, G. Li, K.R. Sollins, and A.K.H. Tung, Effective Keyword-Based Selection of Relational Databases,Proc. ACM SIGMOD Conf., pp. 139-150, 2007.

[6] Q.H. Vu, B.C. Ooi, D. Papadias, and A.K.H. Tung, A Graph Method for Keyword-Based Selection of the Top-K Databases, Proc. ACM SIGMOD Conf., pp. 915-926, 2008.

[7] H. He, H. Wang, J. Yang, and P.S. Yu, Blinks: Ranked Keyword Searches on Graphs, Proc. ACM SIGMOD Conf., pp. 305-316, 2007.

[8] G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou, Ease: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-Structured and Structured Data, Proc. ACM SIGMOD Conf., pp. 903-914, 2008.

[9] Feifei Li, Member, IEEE, Bin Yao, Mingwang Tang, and Marios Hadjieleftheriou, Spatial Approximate String Search, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 6, JUNE 2013.