IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Special Issue, 4, Feb 2019 E-ISSN: 2347-2693

Index of Kumbakonam Using Top-K Query Retrieval Algorithm

M. Nagalakshmi[1*], M. Vaishnavi[2]

[1]M.Sc Computer Science, Idhaya College for Women, Kumbakonam, Tamilnadu, India
[2]Department of Computer Science, Idhaya College for Women, Kumbakonam, Tamilnadu, India

Corresponding Author: mvaishu89@gmail.com

Available online at: www.ijcseonline.org

***Abstract*—**Top-k denotes to the method which only returns the top K most important objects according to a given ranking function. To tackle the limitations of the existing Top-k query, we proposed a modified Top-k query algorithm. In this algorithm, we select the data elements which have higher ranking scores on each attribute, and then run a threshold controlling scheme on these data elements.This system reduces the manual and paper work. This proposed system will help the user to know the exact information and details of the facility that they are finding. This system is very much useful for all users. In the proposed system, the user can see the information's of the facilities that are available in the various areas. The user can also see the top ten facilities that are available. And it is used to book travels ticket. And it's contain the search button and corresponding textbox to search the particular information when the user click the search button it will be redirect to the related pages. The admin can add the additional information about the indexes of Kumbakonam. This system provides the addresses and information of the various facilities.

***Keywords*—**Massive data retrieval, Top-k query, I/O debugging, Accuracy.

I. INTRODUCTION

According to statistical information, data is increasingly becoming an important resource in our daily lives. Additionally, the total number of data which the world created in 2010, stored and replicated has reached 1.2ZB. Afterwards, in the year of 2011, the number reaches 1.8ZB. Furthermore, the number has exceeded nearly 8ZB in the year of 2015[1][2]. Under this background, the most important problem we should face is how to reduce the cost of data management and retrieval [3]. Currently, the scale of storage nodes in the modern data center is ranging from tens of thousands to hundreds of thousands. Disk and storage node failures happen more frequently, on the other hand, users require more effective data retrieval system[4][5]. To satisfy the expanding requirement of reliability, availability, and other relevant characteristics of data retrieval, it needs new theories and algorithms to tackle with the current situation. How to design a massive data retrieval system with low time cost and high reliability has been a great challenge[6]. In recent years, with the development of information technology, particularly the emerging of modern network technology, the requirements to collect, memorize and transfer data have been promoted significantly [7]. However, different from the rapid growth of data, the requirements for retrieve useful information from the massive database are not satisfied yet. Information retrieval technology is an important way to tackle this problem. To be useful for real world applications, high performance massive data retrieval algorithms and related software platforms are more and more important [8][9]. Therefore, in this paper, we concentrate on how to effectively search useful information in massive database with high efficiency.

II. METHODOLOGY

It is aimed at providing information about important places of interest. Not only can visitors read and know more about these places, they can also see their relative positioning on Google maps and experience the beauty of India through a large number of photographs from Flicker services. It also suggests travel plans based on users preferences. To make touring more exciting visitors can also share their experiences through the forums on the web site. Another feature of this application is tourists can estimate the travel's budget by only inputting maximum budget of travel in this application. When processing reservation code, an inter-server communication process will occur, main server sends the public key to the hotel server.

Algorithm 1: The modified Top-k query algorithm.
Input: *m* list files (,_1,2, , _ *j f j*∃ _ *m*)
Output: Final query result *R*
(1)Let *id I* to 0
(2) Let number of data elements (*N*) which can be accessed be 0

(3) For *j* 1 to *p*
(4) For *i* 1 to *q*
(5) Obtaining the data element's identifier and *j A*
(6) If *id I* is equal to zero
(7) *N N*)1
(8) Updating *T.id* and *.j T A*
(9)End if
(10) End for
(11) End for
(12) For *j* 1 to *p*
(13) Checking all the elements in *id I*
(14) If *id I* is equal to 1
(15) Updating this attribute to *T*
(16) End if
(17) End for
(18) Return *R*

Utilizing the modified Top-k query algorithm, the proposed algorithm can obtain the data retrieval results. In the next section, experimental results are provided.

III. RESULT AND DISCUSSION

To make performance evaluation, in this experiment, we will compare the performance of our algorithm with the PDG method [14], which is an efficient indexing structure to answer top-K queries using pareto-based dominant graph. Particularly, hardware platform used in this experiment contains Intel 4 processor with 3.0GHz and 8G memory, and the operating system is Windows 8 (64bit). The experimental dataset is collected from the 68- dimension UScensus [15], and the first five dimensions of this dataset is used to construct our own dataset. To test if our proposed algorithm can solve the massive data retrieval, 40 four hundred thousand records are included.

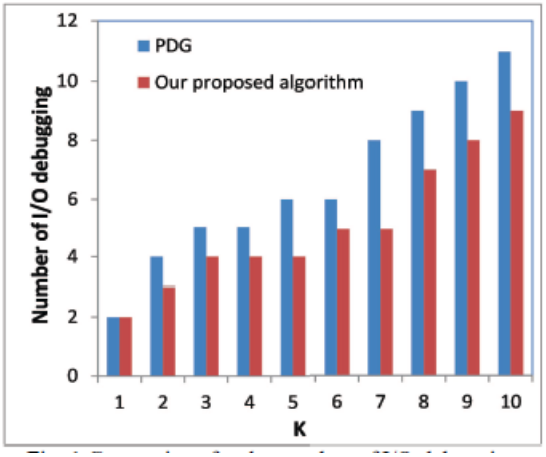

Fig. 1 Comparison for the number of I/O debugging

Fig. 1 shows that compared with PDG method, our algorithm can effectively reduce the number of I/O debugging.

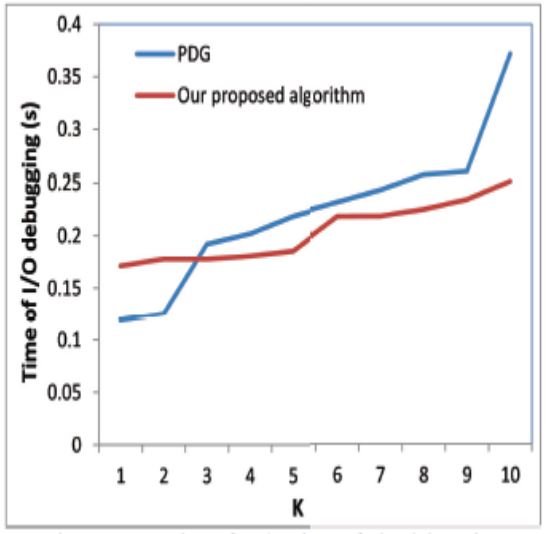

Fig. 2 Comparison for the time of I/O debugging

Fig. 2 demonstrates that when the value of K is smaller than 2, time of I/O debugging of our algorithm is longer than PDG, however, when K is larger than 3, our algorithm can effectively reduce the debugging time. The reason lies in that the proposed algorithm needs much more records

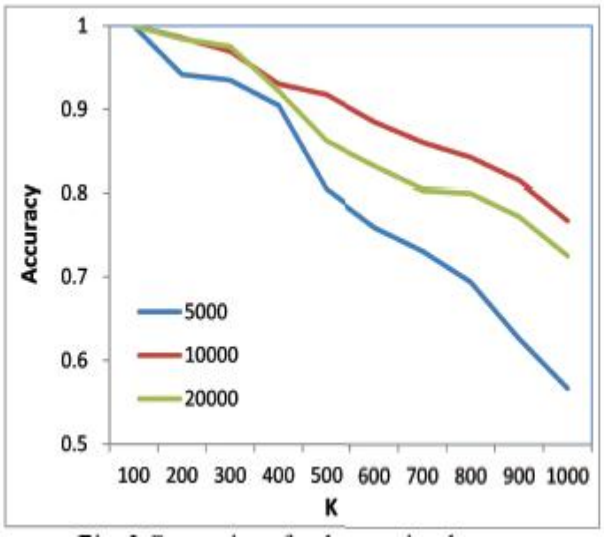

Fig. 3 Comparison for data retrieval accuracy

Fig. 3 illustrates that if K is controlled in the range [1,100], the retrieval accuracy is very high, however, when the value of K increases, retrieval accuracy decreases as well. Because when the number of K is increasing, number of the first level records obviously drops. Integrating the above experimental results, the proposed algorithm can effectively retrieve massive data with high accuracy.

IV. CONCLUSION

This paper proposes a novel massive data retrieval algorithm based on a modified Top-k query algorithm. In particular, the main innovations of this paper lie in that we choose the data elements that have higher ranking scores on each attribute, and then execute a threshold controlling scheme on these data elements. In the end, experimental results verify that our method performs better than existing method in both data retrieval effectiveness and data retrieval accuracy. In the future, we will try to extend our work to the massive multimedia retrieval, such as image retrieval, music retrieval and video retrieval. Furthermore, we will also try to modify our algorithm to parallel computing mode.

REFERENCES

[1] Vega J., Murari A., Pereira A., Portas A., Ratta G. A., Castro R., Overview of intelligent data retrieval methods for waveforms and images in massive fusion databases, Fusion Engineering and Design, 2009, 84(7-11): 1916-1919.
[2] Vega J., Pereira, A., Portas A., et al., Data mining technique for fast retrieval of similar waveforms in Fusion massive databases, Fusion Engineering and Design, 2008, 83(1): 132-139.
[3] Czyzewski A., Bratoszewski P., Ciarkowski A., et al., Massive surveillance data processing with supercomputing cluster, Information Sciences, 2015, 296: 322-344.
[4] Neff Lucas P., Cannon Jeremy W., Morrison Jonathan J., Edwards Mary J., Spinella Philip C., Borgman Matthew A., Clearly defining pediatric massive transfusion: Cutting through the fog and friction with combat data, Journal of Trauma and Acute Care Surgery, 2015, 78(1): 22-28.
[5] Khan Mukhtaj, Ashton Phillip M., Li Maozhen, Taylor, Gareth A., PisicaIoana, Liu Junyong, Parallel Detrended Fluctuation Analysis for Fast Event Detection on Massive PMU Data, IEEE Transactions on Smart Grid, 2015, 6(1): 360-368.
[6] Wang Debby D., Zhou Weiqiang, Yan Hong, Mining of proteinprotein interfacial residues from massive protein sequential and spatial data, Fuzzy Sets and Systems, 2015, 258: 101-116.
[7] Fang Cheng, Liu Jun, Lei Zhenming, Parallelized User Clicks Recognition from Massive HTTP Data Based on Dependency Graph Model, China Communications, 2014, 11(12): 13-25.
[8] Yu Ce, Wang Runtao, Xiao Jian, Sun Jizhou, High Performance Indexing for Massive Audio Fingerprint Data, IEEE Transactions on Consumer Electronics, 2014, 6(4): 690-695.
[9] Gog Simon, Petri Matthias, Optimized succinct data structures for massive data, Software-practice & Experience, 2014, 44(11): 1287-1314.
[10] Liu Dexi, Novel Semantics of the Top-k Queries on Uncertainly Fused Multi-Sensory, Journal of Information Science and Engineering, 2015, 31(1): 179-205.
[11] Guo Long, Shao Jie, AungHtooHtet, Tan Kian-Lee, Efficient continuous top-k spatial keyword queries on road networks, GEOINFORMATICA, 2015, 19(1): 29-60.
[12] DimitriouAggeliki, Theodoratos Dimitri, Sellis, Timos, Top-k-size keyword search on tree structured data, Information Systems, 2015, 47: 178-193.
[13] Rahul Saladi, Janardan Ravi, A General Technique for Top-k Geometric Intersection Query Problems, IEEE Transactions on Knowledge and Data Engineering, 2014, 26(12): 2859-2871.
[14] Zou Lei Chen Lei, Pareto-based dominant graph: An efficient indexing structure to answer top-K queries, IEEE Transactions on Knowledge and Data Engineering, 2011, 23(5): 727-741.
[15] DeNavas-Walt Carmen, Proctor Bernadette D, Smith Jessica C, US Census Bureau, current population reports, Income, poverty, and health insurance coverage in the United States, 2008, 60-236.

