IMPROVING THE PERFORMANCE OF CONTINUOUS TOP-K MONITORING ON DOCUMENT STREAMS

S.kannan[1], T.Meyyappan [2], SM.Thamarai[3]

*[1]Departmentmet of Computer Science, Alagappa University,Karaikudi, Tamilnadu, India. kannanghss@gmail.com[1]
*[2]Professor Department of Computer Science, Alagappa University, Karaikudi, Tamilnadu, India.meyyappant@alagappauniversity.ac.in[2]
*[3] Guest Lecturer, Alagappa Government Arts College,Karaikudi, Tamilnadu, India. @yahoo.co.in[3]

Abstract— The processing of document streams performs a vital role in many records filtering structures. emerging programs. It includes information replace filtering and social community notifications, demand providing end-customers with the most applicable content material to their alternatives. In this work, consumer choices are indicated through a hard and fast of keywords. An imperative server monitors the file movement and constantly reviews to every user the top-k documents that are most relevant to user keywords. Our objective is to aid large numbers of users and high flow prices, Our answer abandons the traditional frequency-ordered indexing technique. Instead, it follows an identifier-ordering paradigm that suits better the nature of the hassle. While complemented with a singular, domestically adaptive approach, our method offers (i) confirmed optimality w.r.t. the range of queries in line with flow event, and (ii) an order of magnitude with reaction time (i.e., time to refresh the question effects) shorter than the state-of-the-art technology.

*Keywords T*op-k query; Continuous query; Document stream

I INTRODUCTION

Motivation. Given a dataset and a preference function, a top-k query returns k objects with the highest preference function score among all objects. In some applications, the top-k results are to be returned in ranked order based on preference scores, as the higher ranked objects may be more preferred. In the previous hotel room bidding example, although the booking system always monitors the top k bids, at any particular time point, only a subset of these k rooms, say k' ($k' \leq k$) rooms, will be available. The availability of the rooms, indicated by k', changes continuously over time, but will never exceed k. At any particular moment, only the top ranked k' bids with highest price from the travelers can be accepted by the system for booking. Therefore, in such scenarios, the top-k result needs to be returned in the ranked order of the preference scores.

Challenges. Efficient execution of continuous top-k queries in streaming environments is challenging. The techniques developed for top-k queries in conventional databases [4, 8, 11] cannot be directly applied nor easily adapted to fit streaming environments. This is because the key problem they solved is, given huge volumes of static data, how to pre-analyze the data to prepare appropriate meta information to subsequently answer incoming top-k queries efficiently [4, 8]. Streaming data however is dynamic with its characteristics dramatically changing over time. Given the real-time response requirement of streaming applications, relying on static algorithms to re-compute the top-k results from scratch for each window is not feasible in practice [10]. Therefore, the key problem to be tackled for continuous top-k query execution is to design a top-k maintenance mechanism that efficiently updates the top-k results even under extremely high input data rates and over huge query windows. The state-of-the-art technique [10] for this problem did not succeed to eliminate the key performance bottleneck of being forced to periodically recompute the top-k results from scratch. Corresponding to the window sliding process, any incremental top-k maintenance mechanism needs to handle the impact of the insertion of new objects and of the expiration of existing objects to the top-k result set. The major challenge for designing such a mechanism lies in handling expirations, as they may frequently trigger the expensive recomputation process. More specifically, when the existing top-k objects expire from the query window, while the new objects fail to fill all the “openings" left by the expired objects, the recomputation process must now search for the qualified substitutes among all objects in the window. This recomputation process represents a serious bottleneck for top-k maintenance in terms of both CPU and memory consumption. Computationally, when searching for qualified substitutes for new top-k

results, this process has to look at potentially all objects in the query window.

II RELATED WORK

Contains the k tuples with the highest scores according to f. The problem is well-studied in conventional databases but the existing methods are incompatible to highly dynamic environments involving numerous long running queries. K.Mouratidis et.al [3] proposed algorithms for the continuous monitoring of top-k queries To achieve real-time query estimation the suitable tuples are stored in main memory. The valid records are arranged by using grid based index schema. Grid based index preserves a book keeping structure. The top k query is important for several online applications such as communication and sensor networks, stock market trading, and profile based marketing etc.

Top k query evaluation can be performed by using the count based and time based sliding window. The count based window W contains the most recent items and the time based window W contains all tuples that arrived within a fixed time instances. The task of the query processor is to constantly report the top k set of every monitoring query among the valid data. When a query q first arrives at the system, its result is computed by the top-k computation module which searches the minimum number of cells that may contain result records. Two algorithms are used for the continuous evaluation of Top k monitoring. future results by reducing the problem to sky band maintenance over a subset of the valid records. The Top k Monitoring Algorithm consists of three modules such as grid based index structure, top k computation module and maintenance module. The grid based index is represented by using 2-dimensional space,. The grid structure contains cells Each grid cell contains the points. Each point p consist of following attributes where id is the unique identifier, x and y are the attributes and t is the arrival time. The grid based index schema allows the continuous processing of multiple queries [4]. It avoids expensive reorganization costs. It can be broadly classified into regular grid structure and irregular grid structure. The benefit of the regular grid is that insertions and deletions are processed efficiently. It is significant to supply an efficient mechanism for evicting the expiring records. In the count and time based sliding window the tuples are evicted in First In First Out (FIFO) manner .All the records are stored in a single list .The new arrivals are placed at the end of the list. The tuples that fall out of the window are discarded from the head of the list. The running queries q is stored in a query table. Query table maintains for each query q contains a unique identifier q.id,its scoring function q.f, the number of tuples required q.k and its current list q.top_list. The score of the kth point in q.top_list is referred to as q.top_score. To restrict the scope of the top k maintenance algorithms each cell is associated with an influence list ILc .

In computation module the result of a query q is obtained by sorting all the score of the cell c according to the maxscore(c) and processes them in descending order. The search terminates when the cell c under the consideration has maxscore(c) $\leq$ q.top_score (q.top_score is the score of the kth element in the q.top_list) . The operation on the maintenance module occurs after the computation of the initial result. When a new tuples arrives at the system the oldest tuples expires. Let Pins be the set of incoming tuples and Pdel be the set of evicted ones. For each p $\in$ Pins it initially insert into the point list of the corresponding cell c.Then it scan the influence list ILc of c and updates the result of every q $\in$ ILc for which score(p) $\geq$ q.top_score . The expunged point p may be part of the result for some of the queries in ILc . For each query q in ILc , If p $\in$ q.top_list , q is marked as affected ,implying that it result has to be computed from scratch when the processing of Pdel is completed. Skyband Monitoring Algorithm applies the reduction of top k to k-skyband queries in order to avoid computation from scratch when the results expire. The skyband maintenance procedure only handles tuples p with score(p) $\geq$ q.top_score. When such a tuple arrives at the system, it is inserted into q.skyband increasing its cardinality. SMA is expected to be faster than TMA, because it involves less frequent calls to the top k computation module. The space requirements of SMA are higher than the TMA, because it maintains the skyband of each query. TMA recomputed the result from the scratch and the SMA maintains a superset of the current answer in the form of a k skyband **.**

III EXISTING METHODS

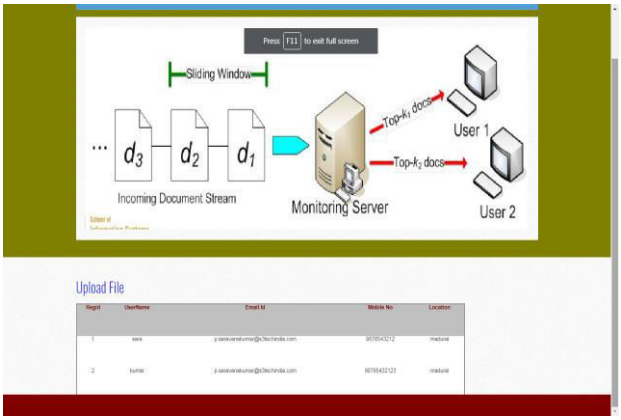
In conventional textual content seek, there are photograph (i.e., one-off) top-okay queries over static record collections. The inverted report is the usual index to organize documents. It comprises a list for each term within the dictionary; the listing for a term holds an access for every report that includes the time period. by sorting the lists in lowering term frequency, and with suitable use of thresholding a photo query can be an-swered by way of processing handiest the top elements of the relevant lists. because of the said sorting, we check with that paradigm as frequency-ordering. This commonplace practice for image queries has been accompanied through most strategies for con-tinuous top-okay search, albeit tailored to the “status” nature of the continuous queries and the surprisingly dynamic characteristics of the record circulation.

Limitation:

- Response time is more
- For updating it takes more time

III PROPOSED SYSTEM

on this, user possibilities are indicated by a hard and fast of keywords. A principal server video display units the record move and continuously reviews to each person the top-okay documents which are



maximum applicable to her keywords. Our goal is to aid massive numbers of users and excessive flow charges, at the same time as nearly instantaneously. Our solution abandons the conventional frequency-ordered indexing approach. rather, it follows an identifier-ordering paradigm that fits higher the character of the hassle. when complemented with a singular, regionally adaptive method, our approach gives (i) validated optimal w.r.t. the range of taken into consideration queries according to movement occasion, and (ii) an order of value shorter reaction time (i.e., time to refresh the question outcomes) than the modern-day trendy

Advantages:

- Response time is less
- For updating it takes less time

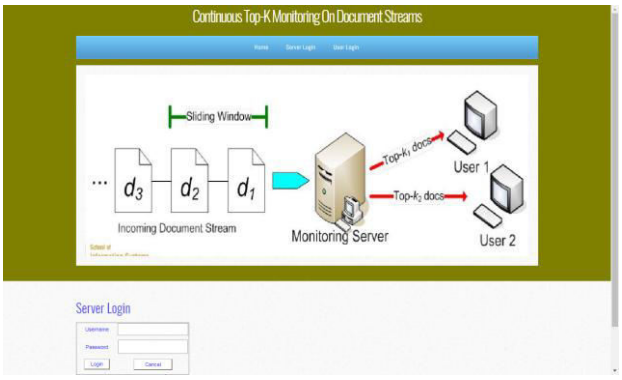
IV IMPLEMENTATION

We assume all valid tuples are sorted in a first-in-firstout list. This provides an efficient mechanism for evicting expired tuples. Newly arriving tuples in each stream are placed at the head of this list and old tuples are dropped from the tail. Note that this is applicable to both countbased and time-based sliding windows. In addition to this list we maintain d sorted lists, one per stream (i.e., for each attribute). Upon receiving hp.id, p.value(i), p.tifrom the i th stream, (p.id, p.value(i)) is inserted in the i th list which is sorted based on the value field. When a tuple expires, it is also removed from the sorted list it belongs to. In order to evaluate a top-k query, the TA algorithm is used. Similar p.value(i), p.tiarrives, p's new score is computed by random access to all other attribute lists. The result list is updated accordingly: if p's score is higher than the least score in this list, p is inserted to the result view. Similarly whenever a tuple expires, the score of its corresponding object decreases. If this object was part of the result view, the result view is updated.

Modules Description:

- Id-ordering techniques
- Minimal reverse id-ordering

VI. EXPERIMENTAL RESULTS



.

Fig: Server Login page

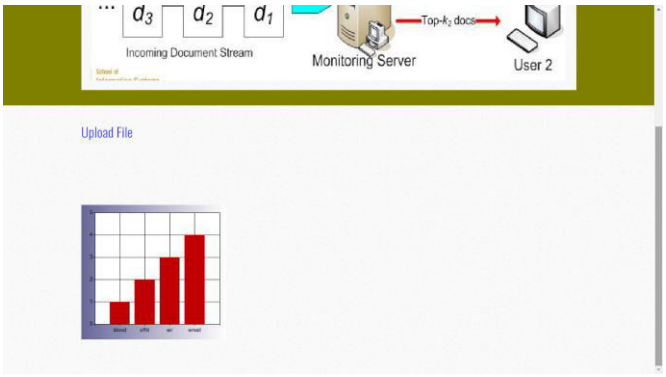


Fig: upload the file from server

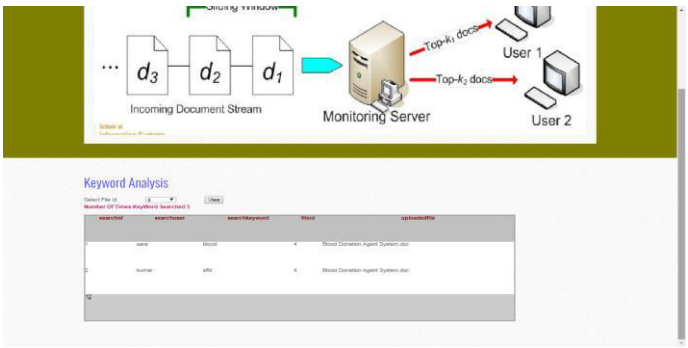


Fig: Upload file analysis screen

V CONCLUSION

In this paper, we proposed a scalable framework for the processing of continuous top-k queries on document streams (CTQDs). A CTQD continuously reports the k-most relevant documents to a set of keywords. CTQDs are employed in many emerging applications, such as email and news filtering. Our

preliminary approach, RIO, adapts the ID-ordering paradigm to the CTQD set-ting. An analysis on RIO reveals that the key factor that determines its performance is the number of iterations it executes.

REFERENCES

[1] W. Kiessling, "Foundations of Preferences in Database Systems,"Proc. 28th Int'l Conf. Very Large Data Bases (VLDB), pp. 311-322,2002.
[2] Y. Tao and D. Papadias, "Maintaining Sliding Window Skylines on Data Streams," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 3, pp. 377-391, Mar. 2006.
[3] K. Mouratidis Management of Data ,pp. 635-646, 2006.
[4] M. Kontaki, A.N. Papadopoulos, and Y. Manolopoulos, "Continuous Top-k Dominating Queries in Subspaces, " Proc. Panhellenic Conf. Informatics (PCI), pp. 675-689, 2008
[5] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "Progressive Skyline Computation in Database Systems" ACM Trans. Database Systems,vol. 30, no. 1, pp. 41-82, 2005.
[6] M.L. Yiu and N. Mamoulis, "Multidimensional Top-k Dominating Queries," VLDB J., vol. 18, pp. 695-718, 2009. [7] B. Babcock, M. Datar, R. Motwani, and J. Widom,"Models and Issues in Data Stream Systems," Proc. ACM SIGMOD Symp. Principles of Database Systems), pp. 1-16, 2002.
[8] V. Hristidis, N. Koudas, and Y. Papakonstantinou, "PREFER: A System for the Efficient Execution of Multi-Parametric Ranked Queries," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 259-270, 2001. [9] M. Kontaki, A.N. Papadopoulos, and Y. Manolopoulos , "Continuous Monitoring of Top k Dominating Queries" IEEE Transactions on Knowledge and Data Engineering, pp, 840 - 853 ,2012.
[10] S. Borzsonyi, D. Kossmann, and K. Stocker, "The Skyline Operator," Proc. Int'l Conf. Data Eng. (ICDE), pp. 421-430, 2001.
[11] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Y. Zien, "Efficient query evaluation using a two-level retrieval process." in CIKM, 2003, pp. 426–434.

S.Kannan, DOB 17.05.1975 obtained B.Sc. (Computer Science)-1993 to 1996,College – Rajapalayalyam Raju's College, Rajapalayam , (Madurai Kamarajar University, Madurai) M.S.c-2001 – 2005 D.D.E - Alagappa University, Karaikudi, B.Ed. (Computer Science) - 2009-2011 D.D.E – Tamilnadu Open University, Chennai , 17 years Working as COMPUTER TEACHER in GOVTHR SEC SCHOOL, Kallamanaickerpatti,, (2001-2002) E.R.R.S.M.Govt.Hr.Sec.School, Alangulam (2002-2008),Govt.Hr.Sec.School,chatrapatti, Virudhunagar District From 2008 to till date.Hospital Management Control - For Padma Hospital Rajapalayam for UG (Front end Tool –Foxpro).Nungambakkam, Chennai (Tools used – HTML, ASP, MS-SQL-Server).

Dr.T.Meyyappan M.Sc,M.Tech.,M. B.A.,M.Phil,Ph.D. currently, Professor, Department of Computer Science, AlagappUniversity, Karaikudi, TamilNadu. He has org anized conferences, workshops at national and international levels. He has published 90 numbers of research papers in National and International journals and conferen ces. He has developed Software packages for Examination, Admission Pr ocessing and official Website of Alagappa University. As a Co-Investigator, he has completed Rs.1 crore project on smart and secure environment fu nded by NTRO, New Delhi. As principal Investigator, he has completed Rs. 4 lakhs project on Privacy Preserving Data Mining funded by U.G.C. New Delhi. He has been honoured with Best Citizens of India Award 2012. His research areas include Operational Research, Digit al Image Processing, Fault Tolerant computing, Network security and Data Mi ning.

SM. Thamarai currently, guest lecturer, Alagappa Government Arts College, Karaikudi, received her Diploma in Electronics and Coomunication Engineering, Department of Technical Education, Tamilnadu in 1989 and her B.C.A. M.Sc. (University First Rank holder and Gold medalist), M.Phil. (First Rank holder) degrees in Computer Science(1998-2005) from Alagappa University. She has published 27 research papers in International, National Journals and conferences. She received her Ph.D. degree in Computer Science in 2014. Her current research interests include Operational Research and Fault Tolerant Computing.