International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-03, Issue 03, Apr 2017

Efficient Filtering Algorithms for Location-Aware Publish/Subscribe

[1]Prof. Sumeet Pate, [2]Pratiksha V. Patil, [3]Prachi M. Jadhav, [4]VinayKumar R.Yadav

[1]Asst. Professor, [2,3,4]UG Student, [1,2,3,4]Computer Engg. Dept. Shivajirao S.Jondhle College of Engineering & Technology, Asangaon, Maharshatra, India

[1]sumeetpate09@gmail.com, [2]pratipatil76@gmail.com, [3]prachijadhav358@gmail.com, [4]vinayy301@gmail.com

Abstract: Location-based services have been widely used in many systems. Existing works employ a pull model or user-initiated model, where a user issues a query to a server which replies with location-aware answers. To provide users with quick replies, a push model or server initiated model is becoming an unavoidable computing model in the next-generation location-based services. In the push model, subscribers register spatio-textual subscriptions to capture their benefits, and publishers post spatio-textual messages. This calls for a high performance location-aware publish/subscribe system to deliver publishers' messages to relevant subscribers. This paper addresses the research challenges that stand up in designing a location-aware publish/subscribe system. This paper proposes an R-tree based index by integrating textual descriptions into R-tree nodes. This paper addresses efficient filtering algorithms and effective pruning techniques to achieve high performance. This paper's method can support both conjunctive queries and ranking queries. This paper implements how to support dynamic updates efficiently. Experimental results show that this method attains high performance which can filter 500 messages in a second for 10 million subscriptions on a commodity computer.

Keywords: Pull Model, Push Model, Efficient Filtering Algorithm.

I. INTRODUCTION

LOCATION-BASED services (LBS) have captured significant attention from both industrial and academic communities. Many LBS services such as Foursquare (foursquare.com) and Google Maps (maps.google.com) have been broadly accepted because these services provide users with location-aware experiences.

The services which are based on location are used in numerous application. In this type of systems any user request for particular service through query to a server and the server gives feedback according to location. For providing this facility user has to enter their interest for finding their need.

Prevailing LBS systems use the pull model or user-initiated model [5], [13], where a user issues a query to a server which then responds with location aware answers. For example, if a mobile user wants to find seafood restaurants nearby, she/he issues a query "seafood restaurant" to an Location Based System, which gives answers based on user's location and keywords.

To stipulate users with instant replies, a push model or server-initiated model is becoming an unavoidable computing model in next-generation location-based services. In the push model, subscribers register spatio-textual subscriptions to capture subscriber's interests, and publishers post spatio-textual messages. This beckons for a high-performance location-aware publish/subscribe system to send the messages to pertinent subscribers. This computing model brings new user experiences to mobile users, and can aid users gain data without clearly issuing a query.

There are numerous existing real-world applications using the location aware publish/subscribe services. The first one is Groupon. Groupon customers register their area of interests with locations and keywords (e.g., "iphone4s" at New York). For each Groupon message (e.g., "iphone4s AT&T package" at Manhattan), the system provider sends the message to the customers who may be interested in the message by examining the spatial proximity and textual relevancy between subscriptions and the message.

The another application is location-aware AdSense, which spreads traditional AdSense (www.google.com/adsense) to support location-aware services. The advertisers register their location-based advertisements (example "seafood" at Manhattan) in the system. The system pushes relevant advertisements to mobile users based on their locations. And also on the contents they are browsing (e.g., web pages).

The next application is tweet delivery. To collect feedback of their products in a specific area from Twitter, market analysts register their interests (e.g., "ipad2" at LA). For each tweet (e.g., "ipad2 is expensive" at LA Airport), the system pushes the tweet to pertinent analysts whose spatio-textual subscriptions [6] match the tweet. The location-aware keyword query yeilds ranked objects that are close to a query location and that have textual descriptions that match query keywords [5].With the proliferation of geo-positioning and geo-tagging [6], spatial web objects that own both a geographical location and a textual description are gaining in prevalence, and spatial keyword queries that exploit both location and textual description are gaining in prominence.

The main challenge in a publish/subscribe system is to gain high performance [12]. A publish/subscribe system should be compatible to tens of millions of subscribers and deliver messages to pertinent subscribers in milliseconds. Since messages and subscriptions comprise both location data and textual description, it is costly to provide messages to pertinent subscribers. This beckons for an efficient filtering technique to support location-aware publish/subscribe services.

To address the challenge, the method is proposed known as a token-based Rtree index structure (called R^t -tree) by integrating each Rtree node, with a set of tokens that will be selected from subscriptions. Using the R^t -tree, this paper develops a filter-and-verification framework to efficiently deliver or provide a message. To lessen the number of tokens associated with R^t-tree nodes, this paper selects some high-quality representative tokens from subscriptions and associate them with R^t -tree nodes. This technique reduces index sizes as well as improves the performance. Experiments on large, real data sets show that this method achieves high performance. making the following contribution. It introduces a new computing model and formalizes the location-aware publish/subscribe problem.

II. LITERATURE SURVEY

There are some existing methods used which are as follows

Keyword-first method: It first uses existing content-based publish/subscribe techniques to generate the candidates that satisfy the textual constraint, e.g, using inverted lists. Then it verifies the candidates to check whether they satisfy the spatial constraint. Obviously this method produces large numbers of candidates and leads to low performance [12].

Spatial-first method: Different from the keyword-first method, it first generates the location-based candidates that satisfy the spatial constraint, using existing methods, e.g., segment tree or R-tree. Then it filters candidates which do not satisfy the textual constraint. This method also produces enormous numbers of candidates and has poor performance.

Spatial keyword search based method: There are several studies on spatial keyword search by using a pull model. In this model, the underlying data are a set of objects with locations and keywords. A user submits a spatial keyword query, and the system returns top-k relevant objects by considering spatial and textual proximity between the query and objects. They incorporate keywords (e.g., signature files or inverted lists) into R-tree nodes.

General Matching Algorithm: By using this algorithm the matching problem can be solved easily by testing an event against each subscription. This algorithm initially pre-processes the set of subscription into a matching tree [2], The second phase match takes internal representation and event, and outputs those subscriptions that match the event.

Extended Bookmark Coloring Algorithm: This is an elegant algorithm for computing PPVs(Personalized page rank vector), to computing the PR(Prestige based relevance)scores [5]. The in-memory BCA algorithm to work in secondary memory is extended first. The graph in large blocks that each exploit the memory available, and do the iterative propagation in a per-block manner are read. The computation stops when the termination condition for the graph is met. A block is likely to be read and written multiple times since it may receive PR scores from other blocks that need to be distributed. Secondly, in BCA which works on unweighed graph for a single preferred object, to support PR score computation for a general preference vector ~uQ on a weighted graph.

III. AIMS & OBJECTIVES

Aims

Though Location-Based services have attracted much significant attention from both industrial and academic communities, these systems use pull model or user-initiated model, where a user issues a query to a server which responds with location-aware answers.

To stipulate users with immediate replies, a push model or server-initiated model is becoming an unavoidable computing model in next-generation location-based services. In the push model, subscribers enroll spatio-textual subscriptions to capture subscriber's interests, and publishers post spatio-textual messages. This beckons for a high-performance location-aware publish/subscribe system to hand over the messages to pertinent subscribers. This computing model brings new user experiences to mobile users, and can aid users gain information without explicitly issuing a query.

The main aim of this paper is to provide the subscriber with the accurate results of their products by pruning some unnecessary product, increase the time in searching the accurate product using the proposed algorithm, ranking the

Table: Comparative Study

Sr.No	Paper Title	Author's Name	Algorithm Used	Problem	Solution	Future Work
1	Matching Events in a Content Based Subscription System	M. K. Aguiler a, R. E. Strom, D. C. Sturma n, M. Astley, and T. D. Chandra.	General Matching Algorithm	Time Complexity is more	Efficient and matching algorithm that uses for speed up constraint query	Develop algorithm more efficient and scalable than other common used matching algorithm
2	Efficient Filtering of XML documents for Selective Dissemination of Information	M. Altinel and M. J. Franklin	Event-based filtering algorithm and several enhancements	Selective dissemination on typically rely on simple keyword matching techniques	Develop several index organizations and search algorithm for large-scale information system.	Develop toolkit for filtering the delivery of data in complex network environment .
3	Retrieving Top K Prestige Based Relevant Spatial Web Objects	X. Cao, G. Cong, and C. S. Jensen	Extended Bookmark-coloring algorithm, Early Stop EBC algorithm, Subgraph based EBC algorithm.	The potential results of such query as being independent when ranking them.	The prestige based relevance to capture both the textual relevance of an object to a query	To provide support for updates and nearby object
4	Collective Spatial Keyword Querying	X. Cao, G. Cong, and C. S. Jensen	Greedy algorithm, Exact algorithm with no index.	Focus on find out individual objects that satisfy a query rather than find out groups of objects where the objects in a group that satisfy a query.	Increasing numbers of objects are present on the web that have an connected geographical area and textual description	Develop approximation algorithms with provable approximation bounds.
5	Location aware publish/ subscribe	G. Li, Y. Wang, T. Wang, and J. Feng	R^t-tree filtering algorithm	It doesn't support both conjunctive queries and ranking queries and speed of searching is low.	An efficient algorithm should be developed to simultaneously prune the results and rank the products.	Develop algorithm to support conjunctive semantics to support ranking semantics and support updates.

product using reviews but the reviews may be fake so there is a need to sort manipulated and non-manipulated reviews.
To address the challenge, the method is proposed known as a token-based R-tree index structure (called R^t -tree) by integrating each R-tree node, with a set of tokens that will be selected from subscriptions. Using the R^t -tree, this paper develops a filter-and-verification framework to efficiently deliver or provide a message. To reduce the number of tokens associated with R^t-tree nodes; this paper selects some high-quality representative tokens from subscriptions and associates them with R^t -tree nodes. This technique reduces index sizes as well as improves the performance.

Objective

1. Spatial Textual Information from the publisher undergoes semantic analysis.
2. Spatial Textual Information from the subscriber undergoes semantic analysis.
3. The information from the database is used to match the spatial textual information.
4. This information then is used for creating R-tree.

It then undergoes the filtering and verification process to provide the users with the most appropriate answers by pruning the results.

IV. PROPOSED SYSTEM

1. To address the challenge, the textual descriptions are integrated into R-tree nodes by an effective index structure R^t-tree.
2. Using the R^t-tree, a filter-and-verification framework is developed.

ISSN : 2454-9150 Vol-03, Issue 03, Apr 2017

3. The number of tokens are reduced which reduces index sizes.
4. This method increases the speed of searching using the specified algorithm.

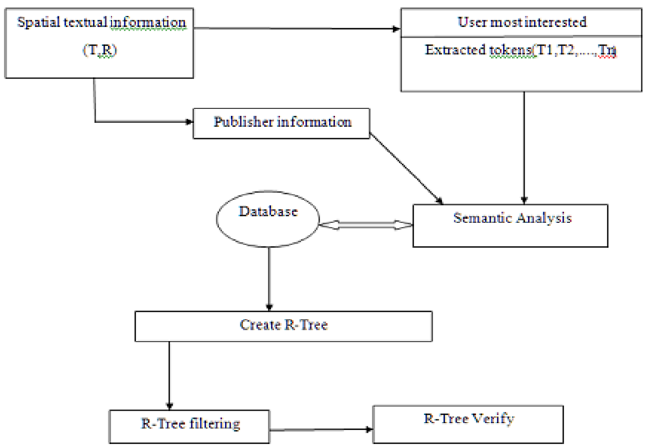

Fig 1.Proposed System

V. ALGORITHMS

R^{t++}-Tree (S, m) Input: S: A subscription set; m: A message

Output: R: Answers of m

1. Build an R^{t++} with root r;
2. Initialize a hash map M ;
3. R^{t++} Tree-Prune (r, m, R,M) ;
<u>Function R^{t++}-Tree-Prune(r, m, R, M)</u>
Input: r: An R^{t++}-tree node; m: A message;R: Answers of m; M: Hash map
Output: R: Answers of m
1. visitFlag = false;
2. for each entry n in node r do
3. if $n.R \cap m.R = \emptyset$ then return;
4. for token $t \in n.T \cap m.T$ do
5. for each subscription s in In[t] do
6. M[s] = M[s] + 1;
7. if !visitFlag& M[s] = i < |s| & n is not a leaf node then
8. visitFlag = true;
9. if M[s] = |s| & $s.R \cap m.R \neq \emptyset$ then
10. R ← s;
11. if visitFlag then
12. R^{t++}-Prune(n,m,R,M)

WORKING: An R^{t++}-tree based algorithm is devised .R^{t++}-TREE first constructs an R^{t++}-tree with root r (line 1) and initializes a hash map *M* (line 2). Then it calls function R^{t++}-TREE-PRUNE to filter message *m*. R^{t++}-TREE-PRUNE first scans each entry (node *n*) from the root. If *n* does not satisfy spatial constraint (line 3), R^{t++}-TREE-PRUNE prunes the node (line 3); otherwise R^{t++}- TREE-PRUNE computes the intersection of its representative-token set and *m.T* (line 4). Then for each token *t* in the intersection, R^{t++}-TREE-PRUNE accesses its inverted list I_n[t] and for each *s* on I_n[t], it increases *M*[*s*] by 1 to count its occurrence number (line 6). If *M*[*s*] = i < |s| and n is not a leaf node, R^{t++}-TREE-PRUNE visits n's children (To avoid repeatedly visiting n's children, visitFlag is set as true in line 8 and if visitFlag is true, visit to such children is done in line 12). If M[s]= |s| and s.R ∩ m.R ≠ ∅, s is an answer and added into the result (line 10).

VI. MATHEMATICAL MODEL

In the proposed system, subscribers register subscriptions to attract their interests. Generally, a subscription *s* includes a textual description *s.T* and spatial information *s.R* which is denoted by *s*=(*T*,*R*). The spatial information is used to attract a subscriber's most interested region. Minimum Bounding Rectangle (MBR) is used to denote a region *s.R*. The textual description is used to capture a subscriber's content-based interests, denoted by a set of token $s.T=\{t1,t2,......,t_{|s.T|}\}$.

A message *m* posted by publisher also contains a textual description *m.T* and spatial information *m.R* denoted by *m*=(*T*,*R*) which respectively have the same meaning as that of subscription.

Let $S=\{s_1,s_{2...},s_{|S|}\}$ denote the set of subscriptions. Given a subscription $s_i \in S$ and a message *m*, a location-aware publish/subscribe system delivers *m* to s_i (s_i is called an answer of *m*), if they satisfy

1) Spatial Constraint: Message *m* and subscription s_i have spatial overlap (i.e.,$s_i.R \cap m.R \neq \emptyset$) and;

2) Textual Constraint: All tokens in subscription *si* are contained in message *m* (i.e.,$s_i.T \subseteq m.T$).

1. Space Complexity of R^t-tree

- Let the height of the R^{t+}-Tree be *H* and the average number of tokens in a subscriptions is $S_{avg.}$ Each token of a subscription is stored at most H times. Thus the token-set complexity is $O(H \times S_{avg} \times |S|)$. There at most $\frac{|S|}{b} + \frac{|S|}{b^2} + \ldots\ldots + \frac{|S|}{b^H} = |S| \times \frac{\frac{1}{b} - \frac{i}{b^{H+1}}}{1-\frac{1}{b}} \approx \frac{1}{b-1} \times |S|$ nodes.
- Thus the space complexity of a node is *O*(*B*) for storing MBRs and child pointers.Thus the overall space complexity is
$$O(\frac{B}{b-1} \times |S| + H \times S_{avg} \times |S|)$$

2. Time complexity of insertion, deletion and updates in R^t – tree

- First corresponding leaf node is located and then the new subscription *s* is inserted into the leaf node. The time complexity of this is *O*(*H*).
- Later, the tokens in the subscription to the TokenSets of the ancestors of the leaf node are inserted. The time complexity of this is $O(H \times |s|)$, where |s| is the number of tokens in subscription *s*.

- For the deletion operation of a subscription, the subscription from the corresponding leaf node is removed with time complexity $O(H)$.
- Consider a TokenSet. For each token in s, the hash value of the token is decreased by 1. If the hash value is 0,the token from the hash value of the TokenSet is removed. The time complexity of this is $O(H \times |s|)^2$.

3. MBR Filter

$$n.R \cap m.R = \emptyset.$$

Where $n.R$ denote MBR,

$m.R$ contains spatial information.

It invalidates spatial constraints under node n have no overlap with m.

4. Token Filter

$$n.T \cap m.T = \emptyset.$$

where $n.T$ denote token sets which can be obtained from the corresponding entry in its parent node,

$m.T$ contains textual description

It invalidates the textual constraint.

5. MBRToken Filter

- To check whether $n.R \cap m.R = \emptyset$, examine whether $n.R$ has a vertex contained in $m.R$ and the time complexity of it is $O(1)$.
- To check whether $n.T \cap m.T = \emptyset$,the hash table of n's TokenSet is used. For each token in $m.T$, if it is contained in the hash table $n.T \cap m.T \neq \emptyset$, is terminated, otherwise the next token in m.T is checked. The time complexity is $O(|m.T|)$

VI. EXPECTED OUTPUT

The publisher registers its product. The subscriber searches for the wished product. By using R^t-tree algorithm the tokens are extracted. The tokens extracted should satisfy the subscriber's wish list. Then the R^{t+}-tree using MBR filters the nodes are pruned.

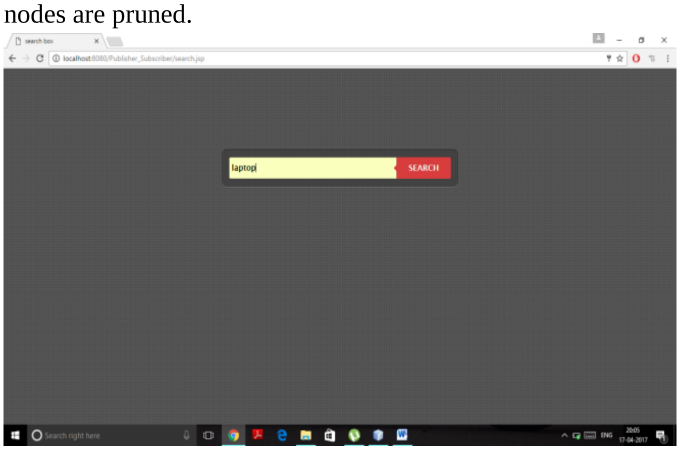

Fig 2: Search Page

Fig 3: R-Tree Page

VII. CONCLUSION

We have tried to implement "Efficient Filtering Algorithms for Location-Aware Publish/Subscribe" by "Minghe Yu, Guoliang Li, Member, IEEE, Ting Wang, Jianhua Feng, and Zhiguo Gong, IEEE transactions on knowledge and data engineering, April 2015" and according to the implementation the conclusion is that an Effective Index structure R^t-tree by integrating textual description into R-tree nodes is proposed. It uses a filter-and-verification framework and devises efficient filtering algorithms. This paper improves performance by reducing the number of tokens in each node which also reduces index size. Algorithms are extended to support both conjunctive and ranking queries. Ranking semantics is also supported.

REFERENCES

[1] Minghe Yu, Guoliang Li, Member, IEEE, Ting Wang, Jianhua Feng, and Zhiguo Gong, "Efficient Filtering Algorithms For Location-Aware Publish/Subscribe".

[2] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra, "Matching events in a content-based subscription system," in Proc. 18th Annu.ACM Symp.Principles Distrib.Comput., 1999, pp. 53–61.

[3] M. Altinel and M. J. Franklin, "Efficient filtering of XML documents for selective dissemination of information," in Proc. 26th Int. Conf. Very Large Data Bases, 2000, pp. 53–64.

[4] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst., 2002, pp. 1–16.

[5] X. Cao, G. Cong, and C. S. Jensen, "Retrieving top-k prestigebased relevant spatial web objects," Proc. VLDB Endowment, vol. 3, no. 1, pp. 373–384, 2010.

[6] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatial keyword querying," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 373–384.

[7] X. Chen, Y. Chen, and F. Rao, "An efficient spatial publish/ subscribe system for intelligent location-based services," in Proc. 2nd Int. Workshop Distrib. Event-Based Syst., 2003, pp. 1–6.

[8] Y.-Y. Chen, T. Suel, and A. Markowetz, "Efficient query processing in geographic web search engines," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2006, pp. 277–288.

[9] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," Proc. VLDB, vol. 2, no. 1, pp. 337–348, 2009.

[10] P. Costa and G. P. Picco, "Semi-probabilistic content-based publish-subscribe," in Proc. 25th IEEE Int. Conf. Distrib. Comput. Syst., 2005, pp. 575–585.

[11] G. Cugola and J. E. M. de Cote, "On introducing location awareness in publish-subscribe middleware," in Proc. IEEE Int. Conf. Distrib.Comput. Syst. Workshops, 2005, pp. 377–382.

[12] G. Li, Y. Wang, T. Wang, and J. Feng, "Location-aware publish/ subscribe," in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp. 802–810.

[13] I. D. Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in Proc. Conf. Data Eng., 2008, pp. 656–665.

