Lamps: Location-Aware Moving Top-k Pub/Sub (Extended abstract)

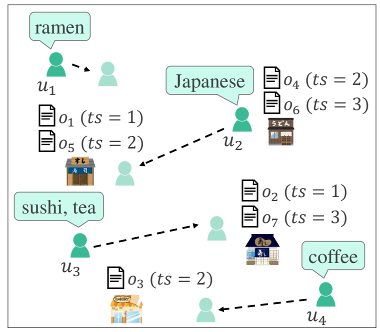
Shunya Nishio, Daichi Amagata, Takahiro Hara
Department of Multimedia Engineering Graduate School of Information Science and Technology Osaka University
shuntdg@gmail.com, {amagata.daichi, hara}@ist.osaka-u.ac.jp

***Abstract*—We propose a novel system, called Lamps (Location-Aware Moving Top-k Pub/Sub), which continuously monitors the top-k most relevant spatio-textual objects for a large number of moving top-k spatio-textual subscriptions simultaneously. Lamps employs the concept of a safe region to monitor top-k results. However, unlike with existing works that assume static objects, top-k result updates may be triggered by newly generated objects. To continuously monitor the top-k results for massive moving subscriptions efficiently, we propose SQ-tree, a novel index based on safe regions, to filter subscriptions whose top-k results do not change. Moreover, to reduce the expensive cost of safe region re-evaluation, we develop a novel approximation technique for safe region construction. Our experimental results on real datasets show that Lamps achieves higher performance than baseline approaches.**

I. Introduction

Because of the prevalence of social networks and GPS-enabled mobile devices, huge amounts of spatio-textual objects have been generated in a streaming fashion. This has led to the popularity of location-aware top-k pub/sub systems [1], [2]. However, these systems assume that the locations of users (subscriptions) are static and cannot support moving subscriptions efficiently. Many real-world applications need to support moving subscriptions, so we propose a novel system, called Lamps, which continuously monitors the top-k most relevant spatio-textual objects for a large number of moving top-k spatio-textual subscriptions simultaneously. Specifically, for each subscription, we score objects based on their spatial and textual similarities, and the top-k results are continuously maintained against streaming objects and the movements of users.

Example 1. *Fig. 1 shows an example of a location-aware moving top-k pub/sub system that delivers e-coupons to potential consumers. In this example, there are four users (subscriptions) and four restaurants (publishers) that continuously generate e-coupons. Each restaurant can generate multiple e-coupons and delete its generated e-coupons. Each user registers his/her interest as a subscription and monitors the top-1 most relevant e-coupon. At timestamp* $ts = 2$*, we assume that five e-coupons (*o_1 - o_5*) have been published and the top-1 result for user* u_3 *is e-coupon* o_1*, because* o_1 *has the highest spatial and textual similarities to the subscription of* u_3*. Also, the top-1 result for* u_2 *is* o_4*. At* $ts = 3$*, new e-coupons are published and each user moves, and then the result for each user is updated. The top-1 result for* u_3 *becomes* o_7*, because* o_7 *is closer to the current location of* u_3 *and its keywords are more similar to the keywords of* u_3*. Furthermore,* u_2 *moves west, thereby the top-1 result for* u_2 *becomes* o_1*.*



o	ts	Keywords
o_1	1	sushi, Japanese
o_2	1	salmon
o_3	2	coffee
o_4	2	Japanese, udon, soba
o_5	2	seafood, beef
o_6	3	udon, tofu
o_7	3	sushi, tea

Newly published.

o : e-coupon
u : user (subscription)
: subscription keywords

Fig. 1: A location-aware moving top-k pub/sub system. At $ts = 3$, o_6 and o_7 are newly published and each user moves to the tip of the arrow.

There are two key challenges for efficiently monitoring the top-k result of each moving subscription. The first challenge is to monitor the up-to-date top-k results for a massive number of subscriptions over a stream of spatio-textual objects. When a new spatio-textual object is generated, for each subscription, a straightforward approach calculates the scores and updates the top-k result. This approach is computationally expensive. The second challenge is to monitor the up-to-date top-k result for each subscription against the movements of users. To monitor the result of a moving subscription efficiently, a safe region technique is often used. (A safe region of a subscription is a region where its top-k result does not change.) Even the state-of-the-art safe region construction algorithm [3] incurs a large evaluation cost if the system maintains many objects. To solve these challenges, Lamps employs a novel index, SQ-tree, which can filter subscriptions whose top-k results do not change when a new object is generated. Furthermore, we develop a novel approximation technique for safe region construction. Our experimental results demonstrate that Lamps achieves high performance.

II. Lamps

Overview. Lamps employs a novel index, SQ-tree, which integrates a Quad-tree with safe regions to effectively maintain moving top-k spatio-textual subscriptions. SQ-tree can filter subscriptions whose top-k results and safe regions do not change when a new object is generated. To reduce the

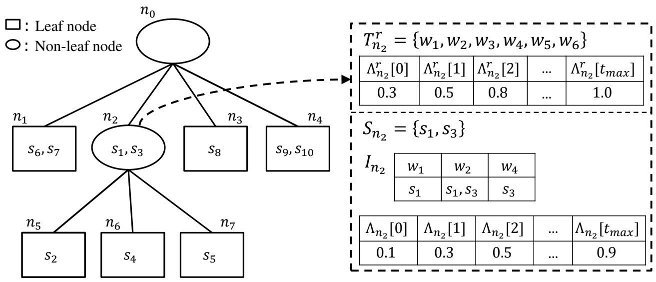


Fig. 2: Illustration of an SQ-tree

expensive safe region evaluation cost, furthermore, we develop a novel approximation technique for safe region construction.

A. SQ-tree

Structure. An SQ-tree maintains subscriptions based on their safe regions and a Quad-tree as shown in Fig. 2. Each node of an SQ-tree maintains the subscriptions whose centroids of safe regions are in the area of the node. Moreover, unlike conventional spatial indices, the SQ-tree maintains subscriptions not only in its leaf nodes but also in its non-leaf nodes. Each node n of SQ-tree stores two thresholds $\Lambda_n[i]$ and $\Lambda^r_n[i]$. Let S_n be a set of subscriptions maintained in n and S^r_n be a set of subscriptions maintained in the subtree rooted at n. $\Lambda_n[i]$ indicates a spatial proximity threshold for subscriptions $s \in S_n$ such that $|s.t \cap o.t| = i$ where $s.t$ and $o.t$ are sets of keywords. Also, $\Lambda^r_n[i]$ indicates a spatial proximity threshold for all subscriptions $s \in S^r_n$ when $|T^r_n \cap o.t| = i$, where T^r_n is a set of keywords held by the subscriptions in S^r_n.

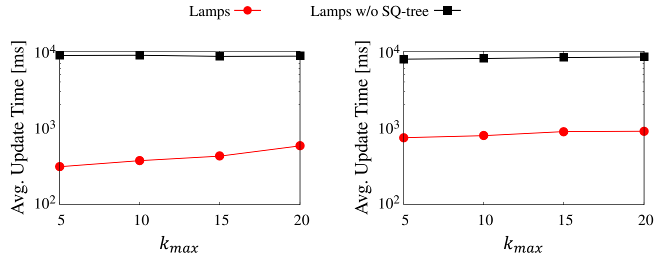
Filtering. SQ-tree can filter a set of subscriptions whose top-k results do not change by a newly generated object o without requesting their current locations. This filtering is performed in two steps. Assume that we check a node n.

Step 1 (Node filtering). Recall that T^r_n is a set of keywords held by $s \in S^r_n$. Given $i_n = \min\{|T^r_n \cap o.t|, t_{max}\}$ where t_{max} is a maximum size of $|s.t|$, we compare $\Lambda^r_n[i_n]$ with the minimum value of the spatial proximity between o and n, i.e., $dist(n, o)$. If $\Lambda^r_n[i_n] < dist(n, o)$, o does not become the top-k result of $s \in S^r_n$. So, we can safely prune n, i.e., we do not need to update the top-k results and safe regions of all subscriptions contained in S^r_n if $\Lambda^r_n[i_n] < dist(n, o)$.

Step 2 (Subscriptions filtering). If n is not filtered by the node filtering, we execute the subscriptions filtering for subscriptions contained in S_n. Let i_{min} be the smallest i such that $\Lambda_n[i] \geq dist(n, o)$. i_{min} indicates the minimum required number of common keywords for o to become the top-k result of each subscription in S_n. That is, o does not become the top-k result of $s \in S_n$ if $|s.t \cap o.t| < i_{min}$. We can safely prune subscriptions s such that $|s.t \cap o.t| < i_{min}$, and we do not need to update their top-k results and safe regions.

B. Approximate safe region

To efficiently compute a safe region for a subscription s, literature [3] proposed a local safe region (LSR). Let O_s be the set of objects for s which is computed based on $\gamma = \frac{score_k(s)}{\alpha} + \Delta$ where $score_k(s)$ is the k-th smallest score of s, α is the preference parameter used in the scoring function, and Δ is a parameter of an approximate ratio. The local safe region construction needs to compute a dominant region for each object $o \in O_s$. Many objects exist in O_s if the system stores a lot of objects, thereby it incurs a large evaluation cost. To avoid this cost, we propose a novel technique that can compute an approximate safe region (ASR), regardless of the distribution of objects. In this technique, we set $\gamma = \frac{score_{k+1}(s)}{\alpha}$. Then O_s becomes empty and we do not need to compute the dominant regions, reducing the evaluation cost.



(a) *TWEETS* (b) *PLACES*

Fig. 3: Impact of k_{max}

III. Empirical study

Because there is no existing index maintaining moving top-k subscriptions, we compared Lamps with the method that accesses all subscriptions and updates the results when an object is generated (denoted by Lamps w/o SQ-tree). We used real datasets (*TWEETS* and *PLACES*) to simulate object stream, and Fig. 3 shows the impact of k_{max}. For each subscription, k was a random integer $\in [1, k_{max}]$. Lamps consistently updates the top-k results faster than Lamps w/o SQ-tree.

IV. Conclusion

Many applications based on pub/sub systems and moving query processing have been attracting a lot of attention. This paper presented an overview of our solution for the problem of monitoring top-k spatio-textual objects for a large number of moving top-k spatio-textual subscriptions. The empirical study shows the efficiency our solution. For a comprehensive coverage, see the full version of this paper [4].

Acknowledgment. This research was partially supported by JSPS Grant-in-Aid for Scientific Research (A) Grant Number 18H04095.

References

[1] L. Chen, G. Cong, X. Cao, and K.-L. Tan, "Temporal spatial-keyword top-k publish/subscribe," in *ICDE*, 2015, pp. 255–266.
[2] X. Wang, W. Zhang, Y. Zhang, X. Lin, and Z. Huang, "Top-k spatial-keyword publish/subscribe over sliding window," *The VLDB Journal*, vol. 26, no. 3, pp. 301–326, 2017.
[3] W. Huang, G. Li, K.-L. Tan, and J. Feng, "Efficient safe-region construction for moving top-k spatial keyword queries," in *CIKM*, 2012, pp. 932–941.
[4] S. Nishio, D. Amagata, and T. Hara, "Lamps: Location-aware moving top-k pub/sub," *TKDE*, vol. 34, no. 1, pp. 352–364, 2022.