Multi-Attribute Top-k Query Processing Leveraging Caching Mechanism in IoT Sensing Network

Zhang Lizhi
School of computer science and Communication Engineering, Jiangsu University

三等奖

Introduction

In the context of edge-cloud collaborative architecture, answering the preference top-k query is a challenging issue. To address this issue, we propose a method to improve the performance of answering numerous continuous preference top-k queries in IoT sensing networks. In this paper, a hierarchical region quadtree is constructed to support efficient query processing by eliminating invalid data at branch nodes. The cloud divides queries into different types according to their preferences, and a responsible edge node caches data records prefetched from the popular blocks for different types of queries based on popularity. The efficient filter thresholds of top-k queries generate from the cache. To further reduce the transmission of invalid data records, a grid index scheme is developed. Experiments indicate our proposed approach is promising in reducing the energy cost of network transmission

Propsed Method

1. Edge-Cloud Collaborative Top-k Query Architecture

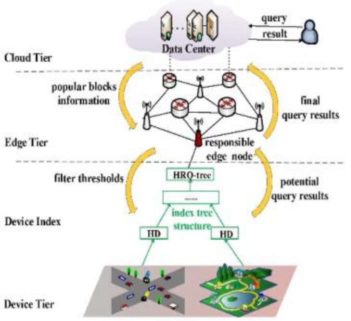


Fig. 1. Overview of three-layer edge-cloud query processing architecture.

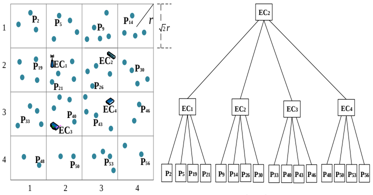
- **Device tier: This tier consists of various smart things in a sensing region. Those smart things periodically collect the multidimensional data of objects, like temperature, vibration.**

- **Edge tier: The edge tier involves edge nodes that have more computational, storage, and energy resources. In Fig. 1, edge nodes take charge of communication and collection of sensory data in the device tier, and the edge node marked red color, is a responsible edge node. The responsible edge node caches data prefetched from popular blocks, disseminates filter thresholds, and aggregates query results.**

- **Cloud tier: The cloud tier is responsible for storing, analyzing, processing the sensory data from the edge tier, holding the global aggregation information from the edge tier. The cloud could acknowledge what attributes users often attach importance to and which grid blocks are popular to generate top-k query results for those queries. The cloud transmits the information of popular grid blocks to the responsible edge node.**

2.Edge Node Caching Mechanism

The region is divided into $2^n \times 2^n$ square grid blocks evenly, where the side-length of each grid block is $\sqrt{2}r$. Then we use the grid block as the elementary unit to construct the HRQ-tree from upper to bottom.



(a) grid block division (b) index tree

Fig. 2. An example of index tree construction.

The popularity of a grid block gb generating top-k results for a type of preference query SQ_i, leveraging the set of grid blocks generating the top-k results for queries in SQ_i in recent s data update iterations:

$$p(gb)_{SQ_i} = \sum_{s=1}^{l} \alpha^s \times val \times \frac{n_q}{|SQ_i|}$$

where $\mathrm{p(gb)}_{\mathrm{SQ_i}}$ is the popularity of a grid block generating top-k results for SQ_i. α represents the attenuation coefficient that is set between 0 and 1. s is the data update iterations and also serves as attenuation exponent for the popularity computation. For the grid block gb at the data update iteration t, if gb generates the top-k result for any query in SQ_i, val is set to 1. Otherwise, val is set to 0. $\mathrm{n_q}$ is the number of queries in SQ_i whose at least one top-k result is produced from gb at the data update iteration t. $|\mathrm{SQ_i}|$ represents the number of queries in SQ_i.

Assume that a type of top-k query SQ_i, the set of cached data records of its popular grid blocks is $\{d_1, d_2, \cdots, d_\lambda\}$ (λ is the total number of data points in popular grid blocks of SQ_i), and the set of data records collected from k_{max} outstanding data points of SQ_i(k_{max} in SQ_i) is $\{d_1, d_2, \cdots, d_{\mathrm{k_{max}}}\}$ Therefore, the set of data records cached for SQ_i in the responsible edge node is $\{d_1, d_2, \cdots, d_{\lambda+k_{max}}\}$ For a top-k query q, $q \in SQ_i$,, assume its top-k result in $\{d_1, d_2, \cdots, d_{\lambda+\mathrm{k_{max}}}\}$ is $\{d_{q1}, d_{q2}, \cdots, d_{qk}\}$ at the current timestamp t, where $sf_q(d_{qi}) \geq sf_q(d_{qj})$, if $i \leq j$.. Then the filter threshold of query q at the current timestamp t can be calculated as:

$$fh_{q,t} = sf_q(d_{qk}) = \sum_{j=1}^{m} q\,.\,\omega_j \cdot d_{qk}.x_j$$

The filter thresholds at each iteration guarantees enough data records to answer top-k queries and does not result in any false negative, but it may cause many false positives when the filter threshold is lower than the actual value.

We construct the grid index for data records aggregating at the parent nodes of the index tree and utilize the dominant relationship between data points to reduce false positives.

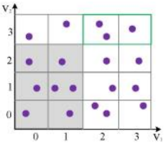
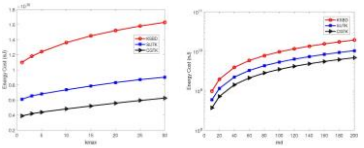


Fig. 3. An example of pruning unnecessary top-k query results. The data points in the gray area are dominated by more than 2 data points. For any top-k query(K<=3),the data points in the gray area are impossible top-k query data records that can be pruned.

Experiment

Results



(a) (b)

Fig. 4. shows the energy consumption of processing top-k queries with different k_{max} and *rnd. CGTK* is our proposed scheme. This result shows that our proposed method reduces the energy consumption of retrieving query results compared to other methods

Conclusion

In this paper, we explore the energy-efficient continuous multidimensional top-k queries processing in edge-cloud IoT sensing networks. First, a hierarchical spatial index tree is constructed to support query processing. Afterward, we develop the edge caching mechanism to produce filter thresholds for queries to identify the potential top-k query results. Finally, the grid index scheme is applied to the internal nodes of the index tree to reduce many false positives. The simulation shows that our proposed method reduces the energy consumption of retrieving query results

References

[1] J. Tang and Z. Zhou, “A priority-aware multidimensional top-k query processing in wireless sensor networks,” Procedia Computer Science, vol. 129, pp. 149–158, 2018.

[2] R. Zhu, B. Wang, X. Yang, B. Zheng, and G. Wang, “Sap: Improving continuous top-k queries over streaming data,” IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 6, pp. 1310–1328, 2017.

[3] H. Jiang, J. Cheng, D. Wang, C. Wang, and G. Tan, “A general frame- work for efficient continuous multidimensional top-k query processing in sensor networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 9, pp. 1668–1680, 2012.

[4] K. Udomlamlert, T. Hara, and S. Nishio, “Subscription-based data aggregation techniques for top-k monitoring queries,” World Wide Web, vol. 20, no. 2, pp. 237–265, 2017.

Contact Information

contacts：Lizhi Zhang
Phone：17826076635
E-mail：2993496492@qq.com