An Efficient Range Search Method Utilizing Detour Routes in Skip Graph

Yuya Miki, Takeshi Kaneko, Ryohei Banno†, and Kazuyuki Shudo
Tokyo Institute of Technology
Tokyo, Japan
miki.y.ad@m.titech.ac.jp

***Abstract*—Skip Graph is a structured overlay that preserves the order of keys and provides efficient routing, especially for complex queries such as range queries. However, existing routing methods for range queries in Skip Graph have problems such as long path lengths and a large number of messages because of inefficiencies. In this paper, we propose a new routing method for range queries named Detouring Range Search (DRS), that provides more efficient routing by utilizing detour routes. In the proposed method, each node that receives a query divides the target range into subranges based on the center of the subordinate node sequence and delegates them to its neighbor nodes. Our evaluation experiments show that the proposed method can reduce the average path length by approximately 20% compared with that of an existing method when the number of nodes is sufficiently large.**

***Index Terms*—Skip Graph, Structured overlay, Peer-to-peer network, Routing algorithm, Range query**

I. Introduction

An overlay network is a logical network built upon an existing network that provides message distribution and a search function for nodes and data. Additional useful functions are provided by constructing a topology independent of the underlying network. In particular, a structured overlay provides high scalability and efficient routing by constructing an autonomous distributed network of nodes following a specific data structure and a protocol. Because of their performance, structured overlays have been applied to a variety of systems, such as distributed file systems and video streaming. They are also being considered for applications in blockchain, which has attracted attention in recent years [1], [2].

In structured overlays, Skip Graph [3] is known for its range query capability. Unlike other structured overlays [4], [5], Skip Graph manages data without hashing keys and can thus construct a topology that preserves the order of the keys. Several routing methods [6]–[8] have been proposed for range queries in Skip Graph, but these methods have problems such as long path lengths and a large number of messages which can lead to high latency and low fault tolerance.

In this paper, we propose a new routing method for range queries in Skip Graph, named Detouring Range Search (DRS). In the proposed method, each node that receives a query divides the target range into subranges based on the center of the subordinate node sequence and delegates them to its neighbor nodes. In this way, the query can be delivered to all nodes in the range with a minimum number of messages and with a short average path length.

This paper is organized as follows. In Section II, we provide an overview of existing routing methods in Skip Graph. Next, we describe the details of the proposed method in Section III, and the evaluation experiments of the proposed method and their results are discussed in Section IV. Finally, in Section V, we present conclusions and future work.

This work was supported by JSPS KAKENHI Grant Number JP21H04872.
† Current affiliation as of December 2021: Kogakuin University

II. Related Work

Skip Graph [3] is a data structure designed based on Skip List [9], where each node belongs to multiple sorted doubly linked lists. Each node has a key and a random string called a membership vector, and the topology is constructed based on these values. A linked list at level i is formed by nodes whose membership vector prefix i digits match.

In this paper, we denote the set of characters that constitute a membership vector as a finite alphabet Σ.

A. Routing method for exact-match queries

For an exact-match query, the search starts at the highest level of the node that issues the query. The query is forwarded by repeatedly hopping within a range that does not exceed the target key and descending one level down. In this way, the higher level acts as a link to shortcut long distances, which enables us to reach the node with a target key with a short path length.

Detouring Skip Graph (DSG) [10] is a method to reduce the path length of routing for exact-match queries by utilizing detour routes. DSG estimates the center of the subordinate node sequence and decides whether to detour based on the value. When estimating the center, it is difficult to obtain the distribution of keys in advance, but evaluation experiments in DSG have shown that in many cases, using the average of two keys is sufficient to reduce the path length. In this way, the path length can be reduced by utilizing detour routes.

DSG is not a routing method for range queries, which is the subject of this paper, but the approach of utilizing detour routes by estimating the center of neighbor nodes in DSG can be applied to range queries.

Proc. 19th IEEE Consumer Communications & Networking Conference (IEEE CCNC 2022), pp.895-898, January 2021

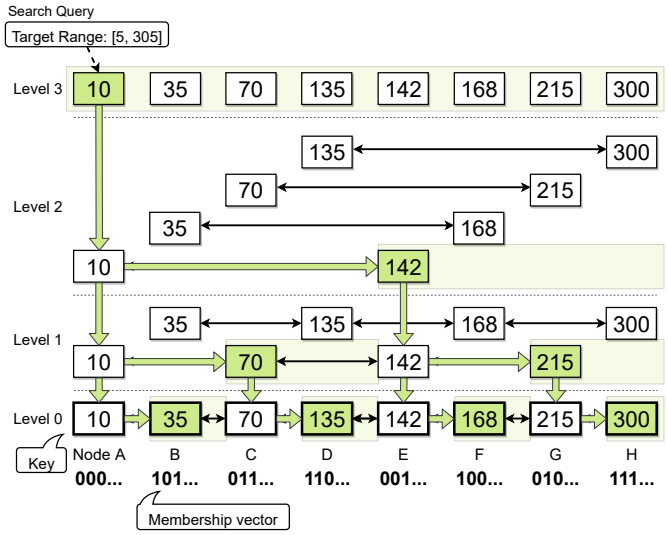

Fig. 1: Example of routing with SFB.

B. Routing method for range queries

For a range query, routing is performed in the same way as exact-match queries until an arbitrary node in the target range receives the query and then starts multicasting within the target range.

Multi-Range Forwarding (MRF) is a routing method for range queries that reduces path length and number of messages by dividing the target range and delegating the subranges [6]. Let us denote the target range of a range query as R. In MRF, each node that receives a query within R divides R into two subranges based on its key. Then, it delegates the subrange to each of the nodes that is a neighbor node at the highest level in the subrange. By repeating this operation for each node that receives a query, all nodes within the target range eventually receive the query.

Since MRF can avoid duplicate transfers, it can handle range queries with a minimum number of messages. However, when using MRF, the path length can be unnecessarily long in some cases; Split-Forward Broadcasting (SFB) is one way to solve this inefficiency [7].

In SFB, the target range is divided into subranges and delegated to neighbor nodes in the same way as in MRF. However, SFB uses a different dividing policy. For a range query whose target range is R, the first node that receives the query within R first divides R into two subranges based on its key. Then, for each of the left and right subranges, it delegates the subrange divided based on the key of a neighbor node rather than its own key. By repeating this operation for each node that receives the query, all nodes within the target range eventually receive the query. Fig. 1 shows an example of routing using SFB where $\Sigma = \{0, 1\}$. The translucent green rectangle represents the target range or subrange, and each node within the range is painted green at the level where it received the query.

However, in some cases, the path length is unnecessarily long because routing with SFB does not fully utilize the links. Specifically, it happens because the query is forwarded in a unidirectional manner, even though the links are bidirectional. To address this inefficiency, we propose a new routing method for range queries. We compare the existing methods with the proposed method in detail in Section IV.

From the perspective of generality, this paper discusses ordinary Skip Graph, which does not have additional links. Therefore, methods that require their own additional links [8] are outside the scope of this paper.

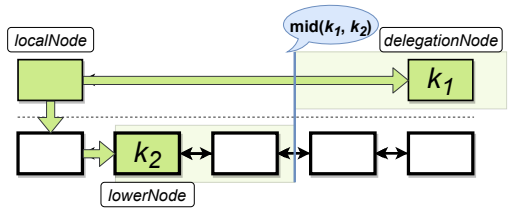

Fig. 2: Dividing position using the proposed method.

III. Proposed Method

We propose a new routing method named Detouring Range Search (DRS), that can reduce the average path length compared to that of SFB, as described in Section II. In the following, we assume that arithmetic operations and absolute values are defined on a set of keys, as in DSG. In the proposed method, the target range is divided into subranges and delegated to neighbor nodes in the same way as in SFB. However, we use a different dividing policy from that of SFB. Specifically, each node that receives a range query estimates the center of the subordinate node sequence, i.e., the center of the node that is adjacent to itself at the highest level and the node that is adjacent to itself at the next highest level, and divides the target range into subranges based on the value. The proposed method with this dividing policy enables nodes to utilize detour routes and make full use of their routing tables and existing links. As for the method of estimating the center, as described in II-A, it is difficult to obtain the distribution of keys in advance, so we use the average of the two keys.

Fig. 2 shows the dividing position for a right subrange in the proposed method. $localNode$ is a node that received a query from another node, or a node that issued a query. $delegationNode$, whose key is k_1, is a neighbor node at the highest level of $localNode$ in the range. $lowerNode$, whose key is k_2, is a neighbor node at the next highest level of $localNode$ in the range. In the proposed method, $localNode$ divides the range at the center of $delegationNode$ and $lowerNode$, although it divides the range at $delegationNode$ in SFB.

Fig. 3 shows an example of routing with the proposed method. For a range query whose target range is R, the first node or $delegationNode$ that receives the query within R or the delegated range first divides R into two subranges by its key. Then, for each of the left and right subranges, it divides the subrange for the center of the subordinate node sequence and delegates them. By repeating these operations,

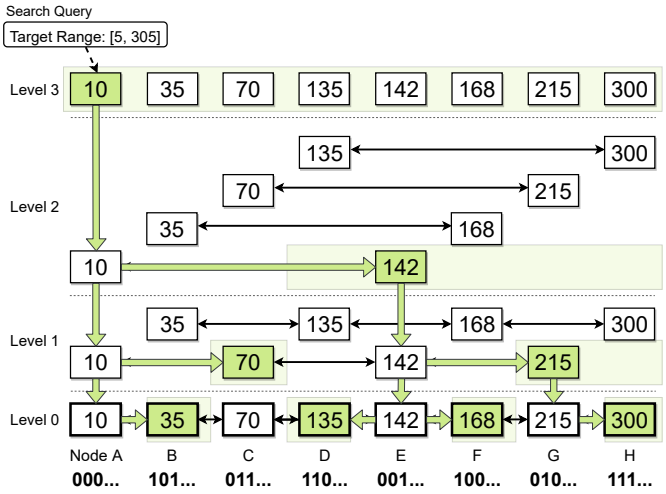

Fig. 3: Example of routing with the proposed method.

all nodes within the target range eventually receive the query. However, since there is no lower level neighbor node at level 0, the subrange is divided by the keys of neighbor nodes and delegated in the same way as in SFB.

Algorithm 1 provides the pseudo code of the proposed method. UPONRECEIVINGATLOCAL is called only once at the node that receives the query within the target range or the delegated range. After that, UPONRECEIVING is called at that node.

In the pseudo code, setRightClosedBound and setLeftClosedBound are functions that change the endpoint of the range to the value specified by the argument and close the right or left side. Similarly, setRightOpenBound and setLeftOpenBound are functions that change the endpoint of the range to the value specified by the argument and open it to the right or left. mid is a function that estimates the center of the two nodes specified by the arguments and returns the key value.

Algorithm 1: Routing process of the proposed method

```
1:  procedure UPONRECEIVINGATLOCAL(range, query)
2:      leftSubRange ← clone of range
3:      leftSubRange.setRightClosedBound(localNode.key)
4:      rightSubRange ← clone of range
5:      rightSubRange.setLeftClosedBound(localNode.key)
6:      UPONRECEIVING(leftSubRange, query, FALSE)
7:      UPONRECEIVING(rightSubRange, query, TRUE)
8:  end procedure
9:  procedure UPONRECEIVING(range, query, isRightSide)
10:     if isRightSide = TRUE then
11:         neighbors ← routingTable.getRightNeighbors()
12:     else
13:         neighbors ← routingTable.getLeftNeighbors()
14:     end if
15:     delegationNode ← neighbors.getDelegationNode(range)
16:     if delegationNode ≠ NULL then
17:         subRange ← clone of range
18:         lowerNode ← neighbors.getLowerNode(range)
19:         if lowerNode ≠ NULL then
20:             divideKey ← mid(delegationNode, lowerNode)
21:         else
22:             divideKey ← delegationNode.key
23:         end if
24:         if isRightSide = TRUE then
25:             subRange.setLeftClosedBound(divideKey)
26:             range.setRightOpenBound(divideKey)
27:         else
28:             subRange.setRightClosedBound(divideKey)
29:             range.setLeftOpenBound(divideKey)
30:         end if
31:         call UPONRECEIVINGATLOCAL(subRange, query) at
            delegationNode
32:         UPONRECEIVING(range, query, isRightSide)
33:     end if
34: end procedure
```

IV. EVALUATION

We conducted simulation experiments to evaluate the path length within the target range and observed the effect of the proposed method on the path length. Keys for the experiments are generated via the following two distributions.

- Uniform distribution
- Power-law distribution

For these methods, we used the same distributions as those used in the experiments in [10].

The details of each experiment are described in the following sections. The alphabet of the membership vector is fixed at $\{0, 1\}$, which is commonly used in the implementation of Skip Graph. Based on the key generated by the designated method, our simulator constructed Skip Graph with 10,000 nodes, and the average path length was measured by changing the number of nodes in the target range N_R for MRF, SFB, and the proposed method. The flow of the measurement is as follows.

1) The leftmost node in the target range issues a range query.
2) Each node forwards the query with the designated routing method while keeping track of the path length from the start node to each node in the target range.
3) After the query is delivered, the simulator calculates the average path length.
4) The above procedure is repeated 100 times for the same topology, and the average value of the 100 trials is calculated. The above procedure is repeated for five different topologies, and the average value is calculated.

The start node, as described in [7], is the first node that receives the query in the range during actual routing. However, since the position in the target range is not always the same, we used the leftmost node in the range as the start node, as in the experiments in [7], to evaluate the performance of the multicast methods in the target range.

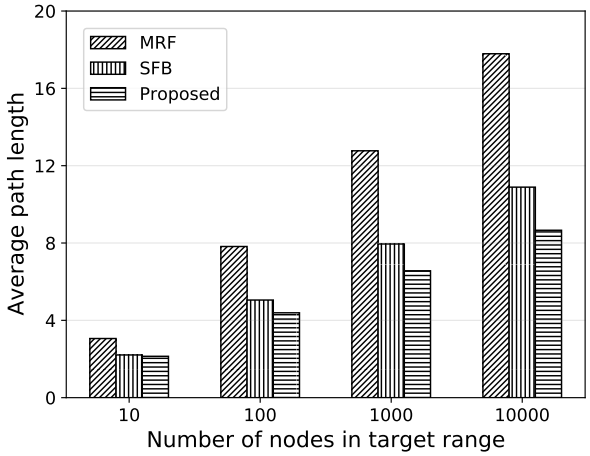

Fig. 4: Comparison of the average path length for a topology where keys were generated by a uniform distribution.

A. Uniform distribution

We generated the keys of the nodes to follow a uniform distribution $P\{v.\text{key} = k\} = \frac{1}{2^{30}} (k \in \{0, 1, ..., 2^{30} - 1\})$ where v.key is the key of node v. The center estimation mid for this distribution is $mid(k_1, k_2) := \frac{k_1+k_2}{2}$.

Fig. 4 shows the average path length measured by the procedure described above. The proposed method has a shorter average path length than that of the other two methods, although the three routing methods show a similar tendency with respect to the change in N_R. In the graph in Fig. 4, compared to SFB, the reduction of the average path length at each N_R is 3.15% ($N_R = 10$), 13.05% ($N_R = 100$), 17.46% ($N_R = 1,000$), and 20.48% ($N_R = 10,000$).

B. Power-law distribution

We generated the keys of the nodes to follow a power-law distribution $P\{v.\text{key} \leq k\} = \int_0^k f(\kappa)d\kappa \, (0 \leq k \leq 2^{30})$ for a probability density function $f(k) = ck^{10} \, (0 \leq k \leq 2^{30})$. Here, c is a constant that satisfies $\int_0^{2^{30}} f(\kappa)d\kappa = 1$. As mentioned in Section II-A, the average of two keys works well as the center of two nodes in many cases, so the mid used in Section IV-A is also used to estimate the center in this key distribution. The purpose of using a power-law distribution is to evaluate the effect of the proposed method on a biased key distribution. For this purpose, we use the topology from Section IV-A with only the key value changed to observe the change in the average path length for two different key distributions.

Table I compares the results measured by the procedure described above with the results in Section IV-A. For two different key distributions, a comparable path length is observed for each of the three routing methods. For the proposed method, the reduction ratio is equivalent to that in Section IV-A for the two routing methods, indicating that the proposed method is effective, even when the key distribution is biased.

TABLE I: Comparison of the average path length for a topology where keys were generated by two different distributions.

		$N_R = 10$	100	$1,000$	$10,000$
MRF	key: uniform	3.06	7.82	12.77	17.79
	key: power	3.06	7.82	12.77	17.79
SFB	key: uniform	2.22	5.06	7.95	10.90
	key: power	2.22	5.06	7.95	10.90
Proposed	key: uniform	2.14	4.40	6.56	8.67
	key: power	2.14	4.41	6.60	8.75

V. Conclusion

In this paper, we proposed a new routing method for range queries in Skip Graph, named Detouring Range Search (DRS). By efficiently using the topology of Skip Graph, the proposed method can deliver a query to all nodes in the target range with a minimum number of messages and a short path length. Evaluation experiments show that the proposed method can reduce the average path length by approximately 20% compared to that of existing methods.

We confirmed that the proposed method is effective even when the key distribution is biased. Our future work will include an analytical evaluation of the path length reduction and an investigation of the center estimation methods.

References

[1] E. Rohrer and F. Tschorsch, "Kadcast: A Structured Approach to Broadcast in Blockchain Networks," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies (AFT'19)*, 2019, pp. 199–213.

[2] R. Banno, Y. Kitagawa, and K. Shudo, "Exploiting semi-structured overlay networks in blockchain systems," *IEICE Communications Express*, 2021.

[3] J. Aspnes and G. Shah, "Skip graphs," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4, pp. 37:1–37:25, 2007.

[4] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Transactions on networking*, vol. 11, no. 1, pp. 17–32, 2003.

[5] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*. Springer, 2001, pp. 329–350.

[6] Y. Ishi, Y. Teranishi, M. Yoshida, S. Takeuchi, S. Shimojo, and S. Nishio, "Range-Key Extension of the Skip Graph," in *2010 IEEE Global Communications Conference (GLOBECOM 2010)*. IEEE, 2010, pp. 1–6.

[7] R. Banno and K. Shudo, "An Efficient Routing Method for Range Queries in Skip Graph," *IEICE TRANSACTIONS on Information and Systems*, vol. E103-D, no. 03, pp. 516–525, 2020.

[8] A. González-Beltrán, P. Milligan, and P. Sage, "Range queries over skip tree graphs," *Computer Communications*, vol. 31, no. 2, pp. 358–374, 2008.

[9] W. Pugh, "Skip lists: a probabilistic alternative to balanced trees," *Communications of the ACM*, vol. 33, no. 6, pp. 668–676, 1990.

[10] T. Kaneko, R. Banno, K. Shudo, K. Abe, and Y. Teranishi, "Detouring Skip Graph: Efficient Routing via Detour Routes on Skip Graph Topology," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1658–1673, 2020.

Proc. 19th IEEE Consumer Communications & Networking Conference
(IEEE CCNC 2022), pp.895-898, January 2021