A circled Bloom filter for the membership identification of multiple sets

Jungwon Lee
Department of Electronic and Electrical Engineering
Ewha Womans University
Seoul, Korea
jungwon0736@ewha.ac.kr

Hyesook Lim
Department of Electronic and Electrical Engineering
Ewha Womans University
Seoul, Korea
hlim@ewha.ac.kr

***Abstract*—A Bloom filter is a simple data structure that identifies the membership of an input against a given set. Various types of Bloom filters have been widely used in recent years. While standard Bloom filters only provide the membership identification of a given set by storing a value of 0 or 1 in a cell, we propose to provide the membership information of multiple sets through a single Bloom filter by storing different values in a cell. In other words, if two different sets are given, the proposed Bloom filter structure allocates 2 bits in one cell, and two different values indicating each set are specified in advance to program the sets. Hence the proposed Bloom filter structure accurately determines the membership of each set and the intersection of two sets by querying a single Bloom filter. In addition, we propose a circled Bloom filter structure to improve the accuracy of the membership identification. Experimental results show that the proposed Bloom filter structure provides the better accuracy with using the half of querying operations compared to two separate Bloom filter structure.**

Keywords—Bloom filter; Double-meaning; intersection; circle;

I. INTRODUCTION

Bloom filters [1] are extensively used in networks as well as in many other areas, and their usage is expected to increase rapidly in the future. The standard Bloom filter is a space-efficient structure by allocating only 1 bit to each cell [2]-[4]. However, as the value stored in the cell of the Bloom filter is 0 or 1, the membership querying result is only true or false, which can only determine whether a given input is an element of the programmed set or not. Therefore, a disadvantage is that the values stored in the Bloom filter cannot be utilized for various purposes. When a number of sets is given, the membership of each set should be determined by using as many Bloom filters as the number of sets. In this paper, we propose a Bloom filter structure that programs multiple sets in a single Bloom filter. We describe the proposed algorithm in case of two sets programmed in one Bloom filter, which can specifically determine the membership of the intersection of two sets as well as the membership of each set. For this purpose, 2 bits are allocated to a cell in a Bloom filter, and each element of two sets is programmed by value 1 or 2 representing the membership of each set, and the elements included in the intersection are programmed as X. The Bloom filter is implemented as a circled structure to provide the wide range of hash indexes.

II. PROPOSED BLOOM FILTER STRUCTURE

Figure 1 shows two sets, S_1 and S_2, and their intersection when the universal set U is given. The proposed Bloom filter can accurately determine set S_1, set S_2, the intersection of two sets, and elements not included in the two sets, by using a single Bloom filter structure.

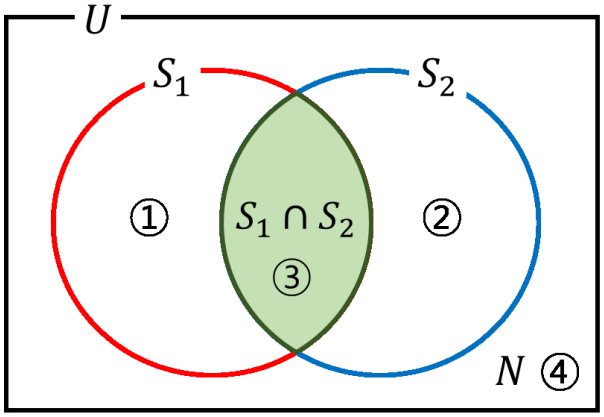

Figure 1. Configuration of set

Unlike the standard bloom filter, 2 bits are allocated for a cell, and the membership of each set is represented by the cell value: 1 represents the element of set S_1, 2 represents the element of set S_2, and 3 (we name X in this paper) represents the element included in both sets.

Assuming that the number of elements in sets S_1 and S_2 is the same, if two separate Bloom filters are implemented for each set, the size of the Bloom filters is $2 * 2^{\lceil \log_2 S \rceil}$. We name this architecture is 2BF. While it is necessary to query both the Bloom filters for the determination of the membership of a given input in the 2BF structure, it is possible to determine which element is included in which set by a single querying in the proposed structure. Additionally, the proposed structure is implemented in a circled shape to improve the accuracy of membership querying. In other words, under the constraint of the same amount of memory usage as the 2BF, the proposed

Bloom filter is programmed by doubling the size of the Bloom filter to create more diverse Bloom filter indexes by the circled shape. Figure 2 shows the proposed Bloom filter structure.

In programming procedure, the elements of set S_1 are firstly programmed by value 1. In programming the elements of set S_2 by value 2, if the cells designated by the hash indexes for an element of S_2 already have value 1, it means that the element is included in the intersection of both sets, and the cell values are converted to value X. In this way, the number of elements programmed to the proposed Bloom filter structure is less than that of 2BF, since the elements included in the intersection are programmed only once, while they are programmed separately to each Bloom filter in the 2BF structure.

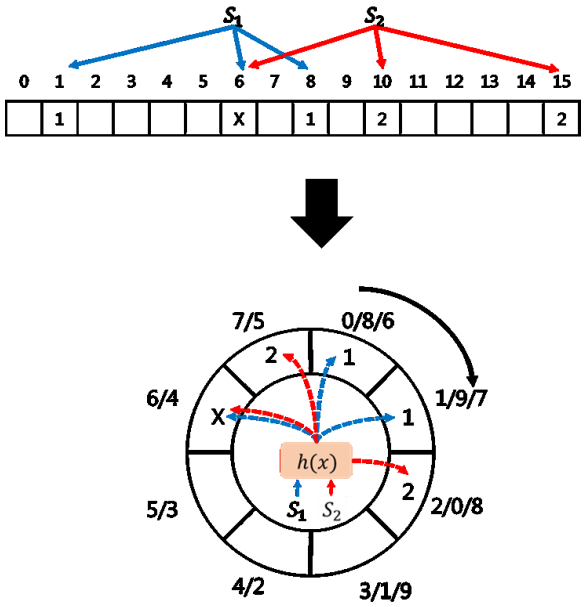

Figure 2. Proposed Bloom filter structure

Querying should be performed for all elements included in the universal set U. Table 1 shows the cases occurred in a querying. If cell values obtained by querying an input are 1 or X, the input is included in S_1. If cell values are 2 or X, the input is included in S_2. If cell values are X only, the input is included in the intersection. On the other hand, if at least a cell value is 0 or if 1, 2, and X are appeared at the same time, the input is determined as negative.

TABLE 1. QUERY RESULT

Query Result	Determination
1 & X	Set S_1
2 & X	Set S_2
All X	Intersection
Any one 0	Negative
1 & 2 & X	Negative

III. PERFORMANCE EVALUATION

We have implemented the proposed structure using C++. The number of universal set U is fixed at 200,000. Each of set S_1 and set S_2 has 100,000 elements. The number of elements in the intersection has experimented with 0%, 10%, and 40% of 200,000. Table 2 shows the number of elements included in each part of Figure 1. In addition, the number of Bloom filter indexes is fixed as 5 for 224KB, 8 for 352KB, and 9 for 416KB. The biggest advantage of the proposed structure is that it performs querying a single Bloom filter instead of querying multiple Bloom filters to determine the membership of each input. Compared to 2BF, the proposed structure requires the half number of querying. If the number of sets is R, the proposed structure requires the $1/R$ number of querying compared to R separate Bloom filter structure.

TABLE 2. NUMBER OF ELEMENTS OF EACH PART BY INTERSECTION RATIO

Intersection ratio	Universal set	①	②	③	④
0%	200000	100000	100000	0	0
10%	200000	80000	80000	20000	20000
40%	200000	20000	20000	80000	80000

Figure 3 shows the accuracy of identifying the membership of S_1 and S_2, when the size of Bloom filter is 416 KB. The accuracy for set S_1 is calculated as (*No. of True Positives for* S_1 + *No. of True Positives for Intersection*) divided by (*No. of Positives for* S_1 + *No. of Positives for Intersection*). The proposed structure has 0.9 ~ 1.4% higher accuracy than 2BF. The accuracy of the proposed structure is improved as the percentage of the intersection increases, since the number of elements programmed to the proposed structure becomes less as the percentage of the intersection increases, while the accuracy of the 2BF is not improved.

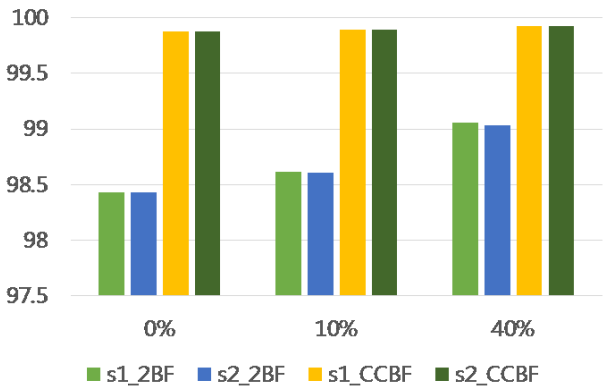

Figure 3. The accuracy of identifying the elements of the set S_1 and the set S_2

Figure 4 shows the accuracy of the intersection according to the size of the Bloom filters. The accuracy of the intersection is calculated as (*No. of True Positives for*

Intersection) / (*No. of Positives for Intersection*). When the size of the Bloom filter is 224 KB, 2BF has only 58% accuracy, but the proposed structure shows approximately 30% higher accuracy.

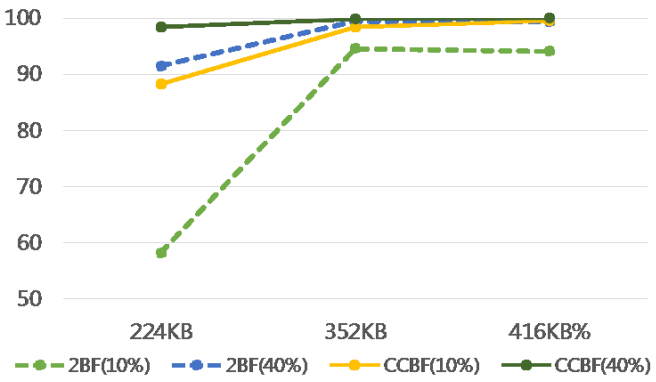

Figure 4. Accuracy of intersection

IV. Conclusion

In this paper, we propose a new Bloom filter structure that improves the accuracy of membership querying. By proposing a circled Bloom filter composed of multi-bit cells to determine the membership of multiple sets, the proposed Bloom filter structure is able to determine the membership of inputs with the $1/R$ number of querying compared to R separate Bloom filter structure when the number of sets is R.

Experimental results show that the proposed structure can provide the membership of set S_1, set S_2, and the intersection 0.5%~1.4% more accurate than the standard Bloom filter under the same memory usage.

Acknowledgment

This research was supported by a National Research Foundation of Korea (NRF), 2018R1A6A3A11040736 and 2017R1A2B4011254.

References

[1] H. Lim, J. Lee, and C. Yim, "Complement Bloom filter for identifying true positiveness of a Bloom filter," IEEE Commun. Lett., vol. 19, no. 11, pp. 1905–1908, Nov. 2015.

[2] [2] H. Lim, K. Lim, N. Lee, and K.-H. Park, "On adding Bloom filters to longest prefix matching algorithms," IEEE Trans. Comput., vol. 63, no. 2, pp. 411–423, Feb. 2014.

[3] [3] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: A scalable wide-area Web cache sharing protocol," IEEE/ACM Trans. Netw., vol. 8, no. 3, pp. 281–293, Jun. 2000.

[4] H. Lim, J. Lee, H. Byun, and C. Yim, "Ternary Bloom Filter Replacing Counting Bloom Filter", IEEE Communications Letters, vol. 21, no.2, pp.278-281, Feb. 2017.