TELKOMNIKA, Vol.15, No.4, December 2017, pp. 1910~1915
ISSN: 1693-6930, accredited **A** by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v15i4.4447

Bloom Filter Implementation in Cache with Low Level of False Positive

Andri Hidayat[1], Fahren Bukhari*[2], Heru Sukoco[3]
[1]Informatic Management Majority, Sambas State Polytechnic, Sambas 79642, Indonesia and Department of Computer Science, Faculty of Mathematics and Natural Sciences,
Bogor Agricultural University, Bogor 16680, Indonesia
[2]Department of Mathematics, Faculty of Mathematics and Natural Sciences,
Bogor Agricultural University, Bogor 16680, Indonesia
[3]Department of Computer Science, Faculty of Mathematics and Natural Sciences,
Bogor Agricultural University, Bogor 16680, Indonesia
*Corresponding author,e-mail: andribise@gmail.com[1], fahren.bukhari@gmail.com[2], hsrkom@ipb.ac.id[3]

Abstract

Searching techniques significantly determine the speed of getting the information or objects. Finding an object in a set is related to membership checking. In the case of massive data, it needs an appropriate technique to search an object accurately and faster. This research implements searching methods, namely Bloom Filter and Sequential Search algorithms, to find objects in a set of data. It aims to improve our system getting a proper item. Due to the possibility of False-Positive existence as a result of Bloom filter technique, there is a potentially inaccurate representation to object sought. Some parameters are influencing False-Positive, namely the number of objects, available bits, and the number of mapped-bit. A Combination of those parameters could decrease the level of False-Positive and improve their accuracy and faster accessibility. In this research, we use three data object variations with the biggest object size of 2000000. Cached objects used in our experiments are between 2 – 20% of variation from the generated objects. The best results with the lowest False-Positive is a combination of bit = 8, mapped bit = 7, and 6% of cache size from 2000000 generated objects.

Keywords: *bloom filter, false positive, sequential search, membership object*

Copyright © 2017 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Nowadays, total internet users/clients increasing has contributed to greater access to a server. This increase can be seen by growing number of people who depend on internet services supporting their activities, even to meet their daily needs. The rise of social media applications and online businesses is one example of interaction widely used by community as well as to support their daily activities.

The high access to data/objects in a server causes the server workload to service requests from clients are also increasing. To address these issues, scalability of a system is needed to handle the demand and increase the amount of data in order not to negatively impact on the performance of the scheme. Excellent scalability of a system can affect the latency and throughput system itself. Latency in this case is that when a number of user increases significantly, then the system provides relatively constant latency. Meanwhile, throughput is that if client significantly increases, the system can improve the ability to handle the number of requests per second linearly. Therefore, we need a method to fulfill client request well.

One of ways to improve response time to client requests is by placing objects closer to the client, it is known as cache [1]. When the client requests an object, the object will be stored in the cache first. This technique is able to serve the demand of an object without taking the object from database as source; therefore, it can reduce communication costs and latency resulted from the process database queries [2]. However, with the limitations of an object that cached, it is possible that client requested object is not in cache. For that, we need to take back the object from the database to give to the client, and this will require a lot more time when the demand is getting higher, and requested objects are in a large set.

To search for objects in a database/very large set, we need searching techniques or methods and related to the checking of object membership in a set [3]. Techniques of object

Received August 25, 2017; Revised November 1, 2017; Accepted November 23, 2017

searching are important to determine speed of the object sought [4]. The checking of object membership in a set can simply be done by checking one by one to the existing object until the object sought is found. This results in high accuracy but less efficient because it takes long time to get an object, especially if the object is in a large set.

Other techniques that can be used are the searching membership techniques of an object by prioritizing speed and tolerates little errors. A Tolerable error is a mistake in the form of objects that are not members but said as a member of the set (false positive) [5]. The technique allowing for a false positive is the Bloom filter technique. Bloom filter is a data structure that is used to determine the membership of an object in a set (Is X ∈ S?) [6]. The performance of a Bloom filter is affected by the number of objects, the number of bits, and the number of hash functions (map bits) used [7, 8].

Both searching techniques use Bloom filter and Sequential Search which has different advantages in getting an object employed as a material for searching memembership of an object in a set and reduce the false positive rate. Therefore, this study will compare the Bloom Filter technique and Sequential Search techniques to get the membership of an object in a large set with the lowest false positive rate until objects obtained. In addition to comparing the techniques Bloom filter with Sequential search, Sequential Search techniques is also used to verify the accuracy of the Bloom filter technique in getting an object in a enormous set.

In order to get the expected results, simulation will be applied in Bloom filter and Sequential search technique. Simulation will use the same data to both Bloom filter and Sequential search.

2. Research Method

The stages of the research activities are illustrated in Figure 1.

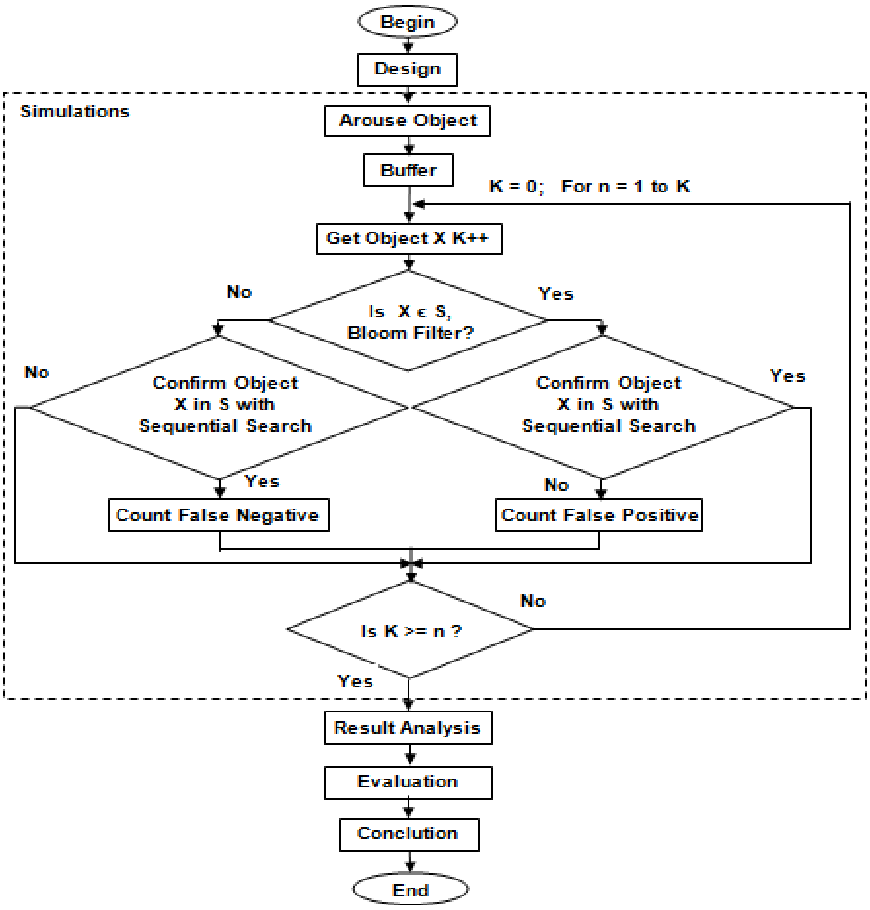

Figure 1. Stages of research

This study begins with the design to create a model of Bloom filters and Sequential search. The next stage is to do the simulation for both designs, Sequential Search and Bloom Filter. Simulations carried out by making the scenario that there is a set of objects with a vast number in it. Furthermore, the steps to determine the half objects of the object set as objects to be stored in the buffer. The next stage or fourth stage is to choose a random object (object x) of the object as a set of objects which will be sought at the buffer. The fifth steps are checking the existence of the object x on the buffer using Bloom filter. The sixth stages are checking the existence of the object x using Sequential search (for your information, that the object x in the sixth step is the same object with the object x that is utilized in the fifth stage). The next stage named count of false positive and the false negative is to calculate the difference between the Bloom filter and Sequential search results. The next stage after simulation is result analysis. This phase will discuss the results obtained from the simulation process. The ninth stage is the evaluation. Evaluation is carried out between the result of Bloom filter and Sequential search. The final stage is the conclusion. It concludes which searching technique is the best, between Bloom filter and Sequential search. Detailes description of each phase carried out in research methods will be discussed in the next sub-discussion.

2.1. Design

Design aims to determine the shape of the design to be created for the Bloom filter and Sequential search and identify the parameters to be used. Algorithms Bloom filter and Sequential search is designed in a form of programming language, models developed is model to search an object in a set which use the same object in the same for both algorithms.

2.2. Simulation

The simulation, in this case, will be divided into several stages, they are generation data/object stage used as the primary input, buffer as a medium container data/objects are cached, Get object x is the choice of objects which will be sought its membership in cache in the buffer. Principal simulation is object searching effort using Bloom filter and Sequential search, as well as counting the number of false positive and false negative of Bloom filter after verification using Sequential search. It also records time used by Bloom filter and Sequential search to get a membership of an object in a set.

2.3. Result Analysis

This phase will discuss the results obtained from the simulation process. Bloom filter method and Sequential search, in this case, would get the same number of objects input. Both of these methods seeks membership of the object which is searched at the same set. All the object that raised will be hashing using CRC32 hashing technique to test the membership of an object in a set using Bloom filter,. After hashing, the object will occupy the bits that are different for each object. This placement varies according to the number of bits specified folder. In this study also uses the number of bits as variables provided to accommodate the object, the number of map bits to place object in bit provided and the variation of the number of objects to be placed in the buffer. For more details, these variables can be seen in Table 1. The possibility of a false positive on a Bloom filter [1] can be expressed by the equation:

$$\left(\frac{1}{2}\right)^k \approx 0.6185^{m/n} \qquad (1)$$

m = number of bits
k = the number of folders bits
n = number of objects

Table 1. Variables Used for Testing

Total Objects	Objects in the buffer (%)	Number of bits supplied	Total of bits folder object mapping
20.000	2	8	2
200.000	4	16	3
2.000.000	6	32	4
	8		5
	10		6
	12		7
	14		8
	16		
	18		
	20		

2.4. Evaluation

This step is performed to evaluate the object searching results that have been obtained using Bloom filter and Sequential search technique. This section will compare the results of object membership using Bloom filter and membership of an object using Sequential search. This evaluation focused on the false positives generated by Bloom filter and the time necessary to obtain the membership of an object by both methods. Since Bloom filter can give false positive results while the Sequential search not, the results of both accuracy and speed of the algorithm will be combined to obtain better results.

3. Results and Analysis

The simulation results that have been done show the influence of variable values in Table 1 for the presence of false positive (FP). It can be seen in Figure 2, which shows significant change to false positive when the number of objects is varied. Figure 2 also shows that the false positive rate to 0 when the number of objects in the buffer is 6% of the actual object 2000000 and stable until the object in buffer is 20%. Furthermore, variations on the number of bits and the map bit although it can lead to false positive changes but it does have great influence on false positives. The influence of the number of bits and the map bit only slightly decreases false positive or insignificant. It can be seen in Figure 3 and Figure 4.

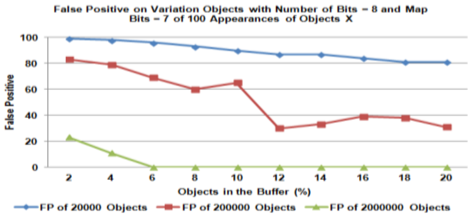

Figure 2. Influence of the Number of Objects to False Positive

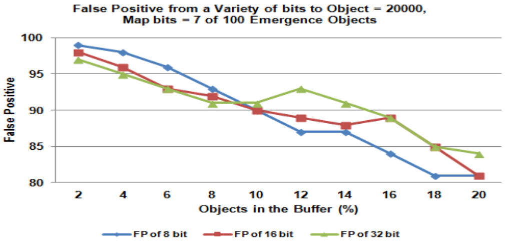

Figure 3. Effect of Number of Bits of the False Positive

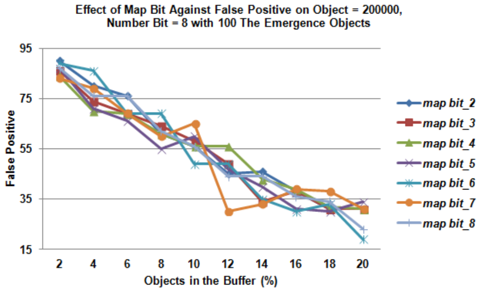

Figure 4. Effect of folder bit against False Positive

In case of time, accessing an object/data using a Bloom filter is faster than using Sequential search. The results of Bloom filter will be compared to the results from Sequential search regarding accuracy and speed. Table 2 shows the comparison of access and speed between Bloom filter and Sequential search.

Table 2 Comparison of Access Time between Bloom Filter and Sequential Search

Objects in the Buffer **(%)**	The Average Time (in µs) of 100 Objects X on a Combination of Appearance:													
	2000000_32_2		**2000000_32_3**		**2000000_32_4**		**2000000_32_5**		**2000000_32_6**		**2000000_32_7**		**2000000_32_8**	
	BF	SS	BF	SS	BF	SS	BF	SS	BF	SS	BF	SS	BF	SS
2	0	220	0	130	0	220	0	150	0	160	10	200	0	230
4	0	240	0	270	0	230	0	290	0	230	0	230	0	200
6	0	250	0	240	0	300	0	220	0	300	0	280	0	230
8	0	270	0	300	0	290	0	340	0	240	10	260	0	200
10	0	280	0	230	0	200	0	220	0	260	0	290	0	300
12	0	420	0	270	0	290	0	270	0	250	0	270	0	290
14	0	280	0	260	0	270	0	250	0	300	0	230	0	330
16	0	240	0	260	0	270	0	290	0	270	0	260	0	220
18	0	180	0	300	0	330	0	280	0	300	0	280	0	250
20	0	270	0	210	10	290	0	290	0	230	0	200	0	270

In general, using Bloom filter is faster than using Sequential search engines, this can be seen from the results of research that combine each of the parameters that can influence the false positive. Searching using Bloom filter is faster than the Sequential search since Bloom filter search space of an object becomes smaller. It is because the utilization of buffer as a temporary space which is determined only by a few percent from existing data sets. Meanwhile, on Sequential search, the search space is equal to the existing set. However, because of the possibility of a false positive on a Bloom filter, it is necessary to reinforce the results of the comparison to the accuracy of the Bloom filter.

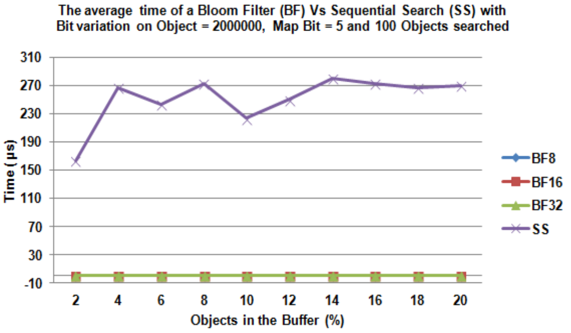

Figure 5. Comparison Charts Bloom Filter Access Time vs. Sequential Search

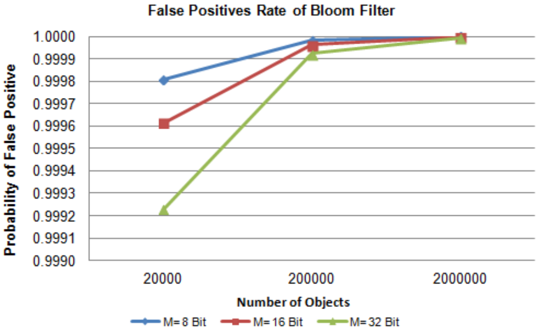

Figure 6. False Positives Rate of the Bloom Filter

Figure 5 shows a comparison of the average time accessing Bloom filter and Sequential search. It can be seen that the time used by Bloom filter to obtain membership of an object is faster (0µs) and constant although the number of objects on the set is continuously increasing. Meanwhile, the use of Sequential search, takes much longer time and is not constant.

From the results that have been to obtained, it can be seen that the best results when object accessing comparing to Bloom filter and Sequential search is on 2000000 number of objects with map Bit by 5 bits for all bit variations available.

Based on the equation (1), with the number of objects that exist and the number of bits provided, false positive level is obtained from the Bloom filter as shown in Figure 6. Thus, it can be seen the high probability of false positives that appeared based on the relationship of object and bit amount available.

3.1. Evaluation

Based on the results, to get the membership of an object in a large set can be used Bloom filter method and Sequential search. Regarding speed, Bloom filter is faster than Sequential search as well as accuracy and precision. Bloom filter still has false positive for some smaller objects but for the larger objects Bloom filter is better. In the condition tested by combination of bit and bitmap, it is found that false positive in Bloom filter can be reduced or even can be non-existent. This reduction can be seen in the test of object X membership in set of objects with combination of 2000000 object in the buffer of 6% of the total number of objects existing, with the bits provided 8 bits and mapped with 7 map bits.

In term of object in a set, out of 100 objects searched, Bloom filter only takes average time 10µs even faster (0µs) compared to Sequential search which takes average time between 10 and 420µs to search with the same number of objects. Even for the larger number of objects, the time required by Sequential search is greater. Bloom filter is more stable at 10µs and even faster (0µs) in search for larger objects such as presented in Table 2.

4. Conclusion

Based on the results and analysis, this study can be concluded that false positives generated from the use of Bloom filter method can be minimized by making the right combinations to the number of objects, the number of bits and mapping the number of objects (map bits). Out of the three parameters affecting false positive on Bloom filter method, the number of object is the most influence parameter to the emergence of false positives. The lowest false positives is obtained with the combination of parameters used on the number of objects 2000000, the number of bits = 8, and the number of map bits = 7. The greater the number of objects in a set, the smaller false positive level obtained. The best number for the availability of objects cached in the buffer is equal to 6-8% of the entire object. Membership searching speed of object in large set using Bloom filter is faster which average time is 10µs and tend to be stable despite of the large number of objects. Therefore, the use of Bloom filter is highly recommended to seek membership object in very large set.

References

[1] Tarkoma S, Rothenberg CE, Lagerspetz E. Theory and Practice of Bloom Filters for Distributed Systems. *IEEE. Commun. Surveys Tuts.* 2012; 14(1) :131–155. doi:10.1109/SURV.2011.031611.00024.

[2] Issa J, Figueira S. Hadoop and Memcached: Performance and Power Characterization and Analysis. *Journal of Cloud Computing a Springer Open Journal*. 2012.

[3] Broder A, Mitzenmacher M. Network Applications of Bloom Filters: A Survey. *Internet Mathematics*. 2004; 1(4): 485–509.

[4] Cheng HY, Ma H. Membership Classification Using Integer Bloom Filter. *IEEE.* 2013: 385-390.

[5] Guo D, Liu Y, Li X, Yang P. False Negative Problem of Counting Bloom Filter. *IEEE Transactions On Knowledge And Data Engineering*. 2010; 22(5): 651-664.

[6] Bloom BH. Space/time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*. 1970; 13(7): 422–426.

[7] Qiao Y, Li T, Chen S. One Memory Access Bloom Filters and Their Generalization. *IEEE INFOCOM*. 2011: 1745-1753.

[8] Qiao Y, Li T, Chen S. Fast Bloom Filters and Their Generalization. *IEEE Transactions On Parallel And Distributed Systems*. 2014; 25(1):93-103. doi:10.1109/TPDS.2013.46.