JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452

Enhanced Adaptive Cuckoo filter searching technique for Approximate Membership Data Structure (AMDS) in network layer.

Mrs.G.Indumathi [1]

ginduavvmspc@gmail.com

Research Scholar

Dr.B.S.E.Zoraida [2]

b.s.e.zoraida@gmail.com

Research Supervisor & Assistant Professor

Received 2022 March 15; **Revised** 2022 April 20; **Accepted** 2022 May 10.

Abstract

In large data center, virtual machines are connected dynamically and data are moved between the physical machines and the virtual machines. With this environment, the Network Layer transfers the packets to the Transport Layer. To improve the security and to increase the speed of the packet transmission in the network layer, packet-filtering is carried out by firewalls based on the decision of routers namely Network addresses, Ports or Protocols. Earlier packet filtering techniques are IP data-gram, Queuing and Scheduling. In order to overcome the limitation in security mechanisms in the above filtering techniques, hashing algorithms are used for analyzing the elements in the packets that are stored in the Hash tables. The hashing algorithms are used in Quotient filter, Bloom filter, Cuckoo filter and Adaptive Cuckoo Filter (ACF). AMDS is termed as An Approximate Membership Data Structures were used to store the fingerprints in cuckoo hash tables. In this paper an attempt is made to improve the searching strategy in Cuckoo filter. Spring constant factor and Leivy Flight theorem is introduced in the searching strategy for best solution identification path. A suit of benchmark functions are employed to verify the performance of network with respect to time, speed and the memory space in the packets transmission. The performance of the Enhanced searching strategy gives better result when compared to ACF.

Keywords: ACF, AMDS, Quotient Filter, Bloom Filter.

1. Introduction

In Network Communication, through the computer networks data has send, then the Internet is divided the data into packets. This packet is consider as a small segment of a huge message. The processes of choosing a smaller segment of your data set to viewing or analysis are called Data Filtering. Packet Filtering is a network security technique that is used to control data flow to and from a network. Then the packets are processed at high speed. At the firewall, packet clarifying is performed to survive whether the packet belongs to the set of members in data structures (i.e.) Approximate set membership data structures (ASMDS) or not.

Based on a specific set of rules the packet clarifiers check the header of the each and every packets to decide whether to accept or drop the packet. In today's Communication channel servers are used in memory , from three to four level it has uncommon of hardware cache ,a pool of disks, several SSDs and a vast pool of DRAM. Under one or more orders of magnitude each successive level of bandwidth, latency and hierarchy were typically increased. Slower medium unnecessarily accessing problem has avoided by the implication of Approximate Set Membership Data Structures (ASMDS) in many application. Suppose an element n is in the set Y in ASMDS as a static variant, An ASMDS like a set data structure to answers a set membership queries (i.e., is an item 'n' an element of the set Y?).

Let an unlike data set, which always reports with reliable data whether n is in Y, an ASMDS is able to report false positives in the data structure set (i.e., falsely state that m is in Y) with an unconditional expected packet error rate is from $0 \leq € \leq 1$. An Approximate Set Membership Data Structures does not report false negatives [29] if an ASMDS reports n not in Y, then it certainly is not. A core benefit of an Approximate Set Membership Data Structures error rate

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452

€ is typically unrestrained data items sizes has encoded, so an ASMDS can often reside more than one levels higher in the memory hierarchy than the slow medium has to requests the filters.

Then the size of the set S is specified ahead of time. However if n in Y is a dynamic variant, the elements of the set are specified one by one, in an on- line network traffic. So it uses trade-off in memory utilization. To reduce optimize the memory with less space requires less time for static variant, it uses the Bloom filter. However in the Bloom filter ,when the element(n) is stored in set Y or when the element(n) is in set Y but it will not in h(n),then it return False positive and no False negative. False Positive is that to check whether the element in a list or not. If it is in set Y,it set bit to 1.Suppose the element is not in a list then it is False negative and set bit to 0 in data structure. In bloom filter deletion cannot be performed.

To perform deletion operation, a new filter is used in member set is Cuckoo filter. However in the Cuckoo filter is a Space-efficient prospect data structures for representing the number of data items in the given data structure set. This Cuckoo filter allows the data to store as fingerprints in approximate membership set of data structures (AMSD), it will perform insertion and deletion operations in AMSD.So it occupies less space of memory with high speed of time for packet processing further query processed in false positive.

In high speed networks during packet transmission (i.e.) when huge amount of elements are processed with duplicates, to avoid duplicates during processing the elements and it occurs waste of time.

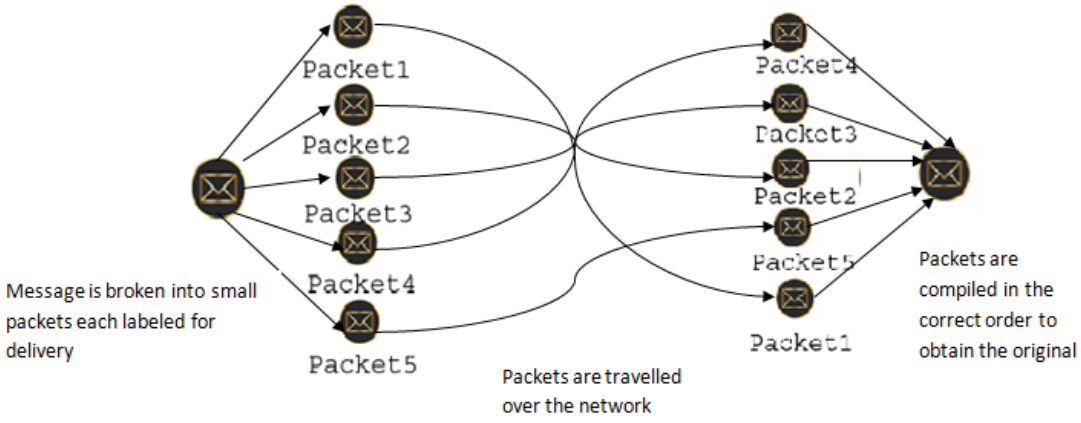

Figure 1: Packet Transmission

To avoid processing duplicates with high probability by testing for membership in the set. These are denoted as false positive rate in the data structure. In the big data field, the Bloom filters are widely-used in the data structure. Hence, It is a very simple powerful data structure to handle a large-scale data by sacrificing a less amount of memory space. Namely, unlike a conventional hash data structure the Bloom Filter performs in that membership set queries with some low probability of false positives. [3].Bloom filter supports only Searching and Insertion not deletion. So switch to another filter is Cuckoo Filter. Let the Cuckoo filter stores the elements fingerprints instead of the set elements storing, it obtained to using an additional hash function for less space, while it still being able to ensure a low false-positive rate probability[4].it perform to be a highest false positive rate to reduce this rate in adaptive cuckoo filter (ACF). It stores the elements as fingerprints in buckets []. It allows fingerprints to some changes, using various operations by using hash functions for better performance than Cuckoo filter. ACF is not a standard replacement for a Bloom filter or a cuckoo filter, but it is an upgrade that should be especially suitable if the previous filter is desired to avoid unnecessary access to unsaved data memory. It stays inside a lot of false negative space. This paper has selected filters to minimize false positives and reduce search time with the Cuckoo Advanced Flexible Filter [EACF].

2. Preliminaries

To explore the static and dynamic variants of the problem, so an attempt is made by using an approximate membership data structure. In ASMDS, the packets are passed by queried elements in the network layer. In the above application the packets are tracked by 5-tuple flows in a link. The 5- tuple consists of Source address, Destination address, Transport

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452
layer protocol identifier, Destination port and Source port, Destination port. Using the 5-tuple flow, it checks whether the x element in approximate membership structure.

2.1 Bloom Filter

A Bloom filter stores the data as fingerprints in data structures. It checks whether the member x in the set or not by the list in membership(5-tuple).It uses space-efficient probabilistic data structure to checking the availability of username of the membership set problem, where the set is in the list of all registered username. Then the False positive means is determined as it might tell that given username is already stored in the list of the membership set. If it return n element value true then it produce False Positive.

Overlay of collaborating and Network of peer-to-peer: The Bloom filters can be used for summarizing content to aid overlay of collaborating and Network of peer-to- peer.

Resource Routing: The Bloom filters can allow the probabilistic algorithms for the locating resources are called as Resource Routing.

Packet Routing: The Bloom filters have to provide a means to speed up or simplify the packet routing protocols are known as Packet Routing.

Measurement: The Bloom filters that provide a useful tool for the measurement of infrastructures which used to create a data summaries in the routers or in the other network devices.

The Bloom filter which supports two operations such as Insert and Search. The Bloom filter allows the false negative and the false positive. A Bloom filter consists of k elements of the hash functions and they may be initially all the bits in an array of the set to be a “0”. If an item has to insert, it be hashes to k positions in the bit array by k hash functions, and then sets all m bits to “1”. Example: p=10, k=3.

2.2 Searching:

The search is processed in the same way, except that you read the corresponding pieces of k in a particular program, if all the bits are searched, then the question should return the truth; otherwise it may be a false return. Bloom filters do not support removal. If Bloom filters can use very little space, but they are not good [6]. Such a false euro, a space-adjusted Bloom filter that uses k = log2 (1 / €) hash functions, moreover only depends on € which may be the size of the data object or the total number of objects.

Theoretical should analyse at least 2 bits (1 / €) for each item, in order to have space for the Bloom filter that places space at the top of the lower theoretical margin. Then the bloom filter does not support removal.

Insertion:

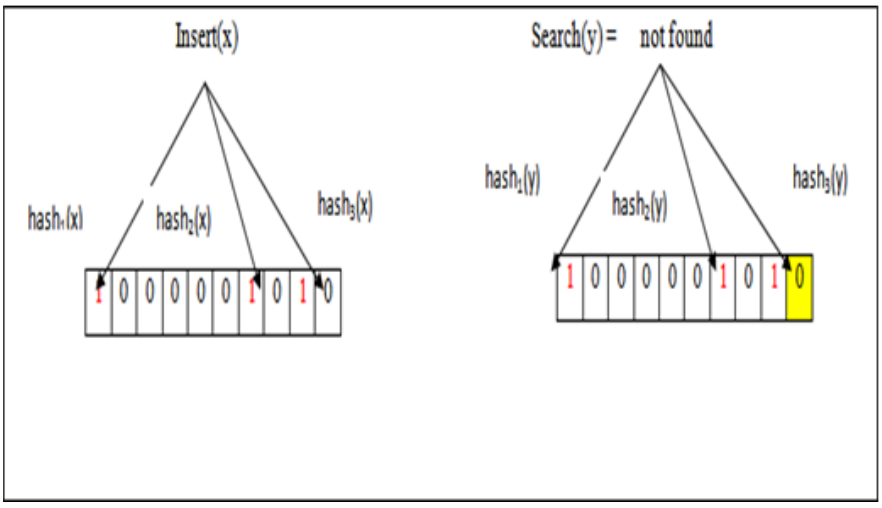

Fig: 2-Hash table methodology

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452

Let a Bloom filter with m bits and the k hashing functions where implies the insertion and membership testing were said as O (k). It indicates that every time when we want to add an element in the set membership or to check the elements available in the set membership Y, if there any element to add, That element must be inserted in to set then performs through the k hash functions and it for both to insert to the set or to check those bits.

2.3 Cuckoo Filter

Based on cuckoo hashing to store the fingerprints of all items in an Approximate Membership Data Structures were implies that to in a cuckoo filter uses a 4 ways set associative hash table. [3]. As it occupies less space and to produce a low False positive probability. Then in the table the two potential buckets are in the given item X required by cuckoo hashing that it is calculated by the following two types of hash functions and let it uses the partial-key cuckoo hashing. Then if the n element is in the set Y f (n), since its two bucket locations will be given by h1 (n) and h1 (n). h2 (f (n)). Hence it supports the Deletion, Search and Insertion.

Search: When we check an element x in set Y, that computes its fingerprint Y (n) and then as well as the bucket locations h1 (n) and h1 (n) .h2 (f (n)). The fingerprints are stored in those buckets are to be compared with Y (n). There if any fingerprints matches with Y (n), then it return as positive or it may be a negative value has to be returned.

h1 (n) =hash (n)

h2 (n) =h1 (n).h2 (f (n))

Insertion: If we Test feature x, which includes two bucket locations and Y (n) fingerprints. If any cell is empty in one of these buckets, we should place Y (n) in the nearest cell or If not, finger f (n) in one of the buckets should be removed, and after that it may be placed repeatedly in the same way as during the cuckoo hash table. This is where we use that both buckets of y can be cut to f (x) and that can be viewed in the bucket where f (x) is removed.

Deletion: When the element (n) in a membership set are to check whether it has to store in a bucket already or may not. The element x is searched for via a lookup. There any if the fingerprint f(x) is found in one of the buckets, then one copy of the fingerprint element has to be removed. Thus the false positive probability of a cuckoo filter can be roughly processed as in the element of hash bucket. If a bits are used for the fingerprints and the element load of the hash table is termed as in the false positive.

2.4 Adaptive Cuckoo Filter

A collection of items in a cuckoo hash table where a flexible cuckoo filter is kept, The flexible cuckoo filter saves as fingerprints for selected items and acts as a cuckoo filter. The hash key function was different compared to the cuckoo filter, and then the cuckoo filter partial key used for cuckoo hashing; The element can be placed in buckets determined by the hash values of the element, as well as those not only on the fingerprints. Flexible filter uses the same hash functions as the cuckoo table hash key, element x and fingerprints are always placed in the corresponding positions. By using a flexible filter, false portfolios emerge when an object not set in S has the same fingerprints as the object stored in one place reached within the search path.

Table 1: Comparison of Filters

Authors	**Filter**	**Search**	**Insertion**	**Deletion**	**False Positive**	**False Negative**
Jehoshua Bruck	Bloom	Agreed	Agreed	Rejection	High	Rejection
Michael D.Mitzenmacher	Cuckoo	Agreed	Agreed	Agreed	High	High
Salvatore Pontarelli	Adaptive Cuckoo	Agreed	Agreed	Agreed	Low	High

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452

Proposed Work	Enhanced Adaptive Cuckoo	Agreed	Agreed	Agreed	Low	Low

It also supports Insertion, Searching and Deletion. The False negative occurs in searching strategy so it takes more time to compute the process. Since it stores the false negative value it occupies more memory space in data structures. To overcome the above problem, some changes are done at searching method in the membership set of fingerprints. Then it uses low space for packet filter (i.e.) Space efficiency and Computation Time is less at Firewalls.

3. Methods

The Cuckoo filter provides that to enables the dynamic deletion and the addition of the items ,where the cuckoo filter can be easily implemented when we compared to the Bloom filter variants with similar capabilities, and towards a similar space constraints, the Cuckoo filter provides lower false positives, it leads to in a particularly at lower capacities.

An Adaptive cuckoo filter where removing the false positives there may occur a significantly lower rate of false positive. The Adaptive cuckoo filter where it similar to the cuckoo filter, it uses a cuckoo hash table to store fingerprints. It allows fingerprint entries to be changed according to the query processed in the hash function and the responses occurs a false positive in a designed manner to minimize the effect on the performance of the filter.

3.1 Construction of Enhanced Cuckoo Filter

In the Proposed work two filters are combined to and reduce the false negative rate in ASMDS and implement an algorithm Leivy flight algorithm, spring constant for low memory space and a reduced amount of time accept in packets

AMDS speed can be tested using Enhanced cuckoo filters for the high speed. EACF also stores the set of fingerprints elements in the set S .The membership set uses the operation of cuckoo hash table. It completely removes false negative with high false positive rate with high performance.

It is a combined strategy of CF & ACF and produce an EACF with high efficiency in an optimized search of nearest one in packet filter. It uses hybrid methodology to prevent from attackers and supports Distributed Denial of Service (DDos) in network applications; also perform Insertion, Search, and Deletion. Here proposed work to use Constant factor and varied factor as variants fmin () & bnest () for the set of S. While deleting the element it won't affect the functions and memory space. It highly evaluate the probability of false positive ,best calls with less time (i.e.) high speed of performance in best solution for the shortest path. The efficiency will be taken from the low bound itself with accurate efficiency classifier in different networks.

3.2 Implementation

Step 1; using the concept of spring constant and Levis theorem in matlab

Step 2: it gives best solution path and reduces the throughput time, minimize the

Memory size in searching methodology.

Searching

Algorithmn1:

//Search the elements in the AMDS Search element n with a single cell

//It will check the elements in buckets

//and perform hash operation Access bucket Ih for I 1 to 4 (x)

f (E) placed on the cell and read Calculate f (n)

In the cell, compare f (E) to f (E) (n)

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452
//if the element in cell it return positive If match then

Return positive Endif

End for

Return negative

Deleting (Remove False rate)

All the elements are stored in ASMDS as bucket and it is denoted by 'x' in a single cell. The elements are stored in hash table h(x).α denotes the operation of reading all elements in the hash table using for loop as fingerprints denoted as fα(x).Then is compared with the set 'E' with the element in the bit. If it matches return positive otherwise negative. Store the new value of α and f α (E) on bucket Ih (E) in the EACF

After Searching strategy perform, it access all the elements in the bucket. Then it read all the elements in bits of the packets. The searched element are stored in E on bucket Ih (E).It performs β=3/2;

$\epsilon=(\gamma(1+\beta)*\sin(pi*\beta/2)/(\gamma((1+\beta)/2)*\beta*2^{((\beta-1)/2)}))^{(1/\beta)}$.Spring constant are used for test functions of each case the elements are searched in the bucket, whether it matches with the new hash table to store false positive and compute the false negative.

There are many tools are available like R, Tool, Weka, Tanagra and Mat lab for the statistical approaches.

4. Results

In proposed work packets are sent from network layer to transport layer and the packets are convert into graphical representation using Mat lab.Matlab is used for procedural computing language at a high-performance. It integrates computation, visualization.

At network layer, the packets are transferred to transport layer in between filter is used for searching the packets ASMDS. The Cuckoo filter takes 0.189s for 25200 packets. The ACF takes 0.326s time for 50000 packets. But in EACF takes only

Asd programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is one of the most widely used mathematical computing environments in technical computing. It is an interactive environment that provides high-performance computational routines and an easy-to-use.

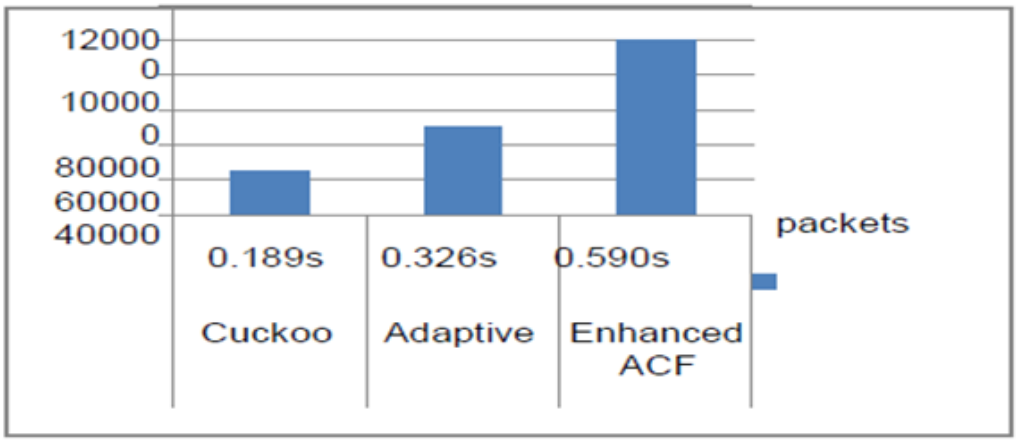

Fig 2:Best Path Way

Table 3.1: Details of searched packets for best solution

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452

Filters	Total packets	Time	Fmin	Deletion Supports
Cuckoo	25200	0.189s	8.5991	Yes
Adaptive	50000	0.326s	1.9502	Yes
Enhanced ACF	100000	0.590s	0.01268	Yes

0.590s for 100000 packets to be searched in Approximate Membership Data Structures. When compared to the above the filters the EACF produce less time, high speed and it occupies less memory space (i.e.) Space efficiency and low Positive Rate .

5. Conclusion

A new data structure, Enhanced Adaptive Cuckoo Filter, is used for the Approximate set membership queries. Many networking problems that were previously solved using Bloom filters and Cuckoo filters can now be solved with EACF. It differs from conventional filters in three ways: (1) support for dynamic deletion of items; (2) improved lookup performance; and (3) improved space efficiency for applications needing low false positive rates (less than 3%).

Using cuckoo hashing techniques, a cuckoo filter can obtain a very high memory occupancy for elements in the ASDMS. As a further key contribution work, we have developed a partial-key cuckoo hashing technique that allows relocation based on only the fingerprint, thus making cuckoo filters substantially more efficient. The present work suggests that the cuckoo filter, which uses buckets of size 4, is likely to meet the needs of a wide range of domains, though the cuckoo filter parameters can be varied according to the application requirements. Considering the future potential for further development and additions to the cuckoo filters, we expect it will further open up new opportunities for their use, but as described-for a fast and efficient way to build packets of data-it already meets the practical requirements for current networks.

References

1. A.Z. Broder, M. Mitzenmacher, Network applications of Bloom filters: A survey, Internet Math. 1 (4) (2003).
2. B.H. Bloom, Space/time trade-offs in hash coding with allowable errors, Commun. ACM 13 (7) (1970) 422–426.
3. H. Cai, P. Ge, J. Wang, Applications of Bloom filters in peer-to-peer systems: Issues and questions, in: NAS'08: Proceedings of the 2008 International Conference on Networking, Architecture, and Storage, Washington, DC, USA, 2008, pp. 97–103.
4. P. Hebden, A. Pearce, Data-centric routing using Bloom filters in wireless sensor networks, in: M. Palaniswami (Ed.), Fourth International Conference on Intelligent Sensing and Information Processing (ICISIP-06), IEEE Press, Bangalore, India, 2006, pp. 72–78.
5. T. Wolf, Data path credentials for high-performance capabilities based networks, in: ANCS'08: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ACM, New York, NY, USA, 2008, pp. 129–130.
6. F. Ye, H. Luo, S. Lu, L. Zhang, S. Member, Statistical en-route filtering of injected false data in sensor networks, in: INFOCOM, 2004, pp.839–850.
7. S. Ratnasamy, A. Ermolinskiy, S. Shenker, Revisiting IP multicast, in: Proceedings of ACM SIGCOMM'06, Pisa, Italy, 2006.
8. P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, P. Nikander, LIPSIN: line speed publish/subscribe inter-networking, in: Proceedings of ACM SIGCOMM'09, Barcelona, Spain, 2009.
9. S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor. Longest prefix matching using Bloom filters. In Proc. ACM SIGCOMM, Karlsruhe, Germany, Aug. 2003.
10. M. Dietzfelbinger and C. Weidling. Balanced allocation and dictionaries with tightly packed constant size bins. Theoretical Computer Science, 380(1):47–68, 2007.
11. B. Fan, D. G. Andersen, and M. Kaminsky. MemC3: Compact and concurrent mem cache with dumber caching and smarter hashing. In Proc. 10th USENIX NSDI, Lombard, IL, Apr. 2013.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 3996-4003
https://publishoa.com
ISSN: 1309-3452

12. L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: A scalable wide-area Web cache sharing protocol. In Proc. ACM SIGCOMM, Vancouver, BC, Canada, Sept. 1998.
13. N. Fountoulakis, M. Khosla, and K. Panagiotou. The multiple orient ability thresholds for random hypergraphs. In Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1222–1236. SIAM, 2011.
14. N. Hua, H. C. Zhao, B. Lin, and J. J. Xu. Rank-Indexed Hashing: A Compact Construction of Bloom Filters and Variants. In Proc. of IEEE Int'l Conf. on Network Protocols (ICNP) 2008, Orlando, Florida, USA, Oct. 2008.