IJIEMR Transactions, online available on 24th Feb 2018. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-02

Title: **A SURVEY ON: ICE BUCKETS: IMPROVED COUNTER ESTIMATION FOR NETWORK MEASUREMENT**

Volume 08, Issue 02, Pages: 74–77.

Paper Authors

MS.G.PRASANTHI, CH.ARAVIND REDDY

Vignan's Lara Institute of Technology & Science

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

A SURVEY ON: ICE BUCKETS: IMPROVED COUNTER ESTIMATION FOR NETWORK MEASUREMENT

MS.G.PRASANTHI[1], CH.ARAVIND REDDY[2]
Assistant Professor[1], Department of M.C.A ,Vignan's Lara Institute of Technology & Science
M.C.A Student[2],Department of M.C.A ,Vignan's Lara Institute of Technology & Science

Abstract:
Measurement capabilities are essential for a variety of network applications, such as load balancing, routing, fairness, and intrusion detection. These capabilities require large counter arrays in order to monitor the traffic of all network flows. While commodity SRAM memories are capable of operating at line speed, they are too small to accommodate large counter arrays. Previous works suggested estimators, which trade precision for reduced space. However, in order to accurately estimfdate the largest counter, these methods compromise the accuracy of the smaller counters. In this paper, we present a closed form representation of the optimal estimation function. We then introduce independent counter estimation buckets, a novel algorithm that improves estimation accuracy for all counters. This is achieved by separating the flows to buckets and configuring the optimal estimation function according to each bucket's counter scale. We prove a tighter upper bound on the relative error and demonstrate an accuracy improvement of up to 57 times on real Internet packet traces.

Introduction

COUNTER arrays are essential in network measurements and accounting.Typically, measurement applications track several million flows [1], [2], and their counters are updated with the arrival of every packet. These capabilities are an important enabling factor for networking algorithms in many fields such as load balancing, routing, fairness, network caching and intrusion detection [3]–[7]. Counter arrays are also used in popular approximate counting sketches such as multi stage filters [8] and count min sketch [9], as well as in network monitoring architectures [10]–[12]. Such architectures are used to collect and analyze statistics from many networking devices [13]. Implementation of counter arrays is particularly challenging due to the requirement to operate at line speed. Although commodity SRAM memories are fast enough for this task, they do not meet the space requirements of modern counter arrays. Implementing a counter array entirely in SRAM is therefore very expensive [14].Counter estimation algorithms use shorter counters, e.g., 12-bits instead of 32-bits, at the cost of a small error. Upon packet arrival, a counter is only incremented with a certain probability that depends on its current value. In order to keep the relative error uniform, small values are incremented with high probability and large ones with low probability. An emphestimation function is used in order to determine these probabilities and estimate the true value of a counter. Estimation functions can be scaled to achieve higher counting capacity at the cost of a larger estimation error. Existing counter estimation techniques suffer from the following problem when facing skewed workloads, as is common in computer networks, a phenomenon known as heavy hitters. Accommodating the counting capacity required by the heavy hitters forces using a large estimation function scale. However, since the heavy hitters often share the same function scale as other counters, the estimation errors for small counters, which correspond to the majority of items, become very large.

Existing system:

Counter arrays are essential in network measurements and accounting. Typically, measurement applications track several million flows and their counters are updated with the arrival of every packet. Counter estimation algorithms use shorter counters, e.g., 12-bits instead of 32-bits, at the cost of a small error. Upon packet arrival, a counter is only incremented with a certain probability that depends on its current value. In order to keep the relative error uniform, small values are incremented with high probability and large ones with low probability. An emphestimation function is used in order to determine these probabilities and estimate the true value of a counter. Estimation functions can be scaled to achieve higher counting capacity at the cost of a larger estimation error. Existing counter estimation techniques suffer from the following problem when facing skewed workloads, as is common in computer networks, a phenomenon known as heavy hitters. Accommodating the counting capacity required by the heavy hitters forces using a large estimation function scale.

Proposed system:

We are the first to present a closed form explicit representation of an optimal estimation function. This enables us to extensively study the various aspects of this function using rigorous mathematical analysis, including the relation between its relative error, memory complexity, estimation symbol range, and even bound the probability of the actual error exceeding a certain value. We present Independent Counter Estimation Buckets (ICE-Buckets), a novel counter estimation technique that reduces the overall error by efficiently utilizing multiple counter scales. We then propose the ICE-buckets technique, which divides counters into buckets, where each bucket is maintained with its own scale parameter, thereby greatly reducing the relative error. ICE-Buckets are also analyzed, and we show a methodological way of configuring its parameters. Finally, we simulate ICE-Buckets using 5 real world traces and compare it to state of the art approaches, demonstrating its substantial benefits.

Modules:

1. Local Upscale: The configuration of the data structure, {wi}B−1 i=0 , is dynamically adjusted to the biggest estimation value in each bucket. Initially, and bucket scales are set to Zero. Whenever a symbol Fij approaches L, we increment wi and upscale Bucket i to use the parameter wi+1. This is done by up scaling all of the flows in bucket i using the symbol upscale procedure. A pseudo code of the local upscale procedure. We note that since the number of counters per bucket S is small, local upscale can be efficiently implemented.

2. Global Upscale: When a counter in a bucket with the maximum scale index (E −1) approaches its maximum value (L − 1), we initiate a global upscale procedure to prevent overflow. The procedure doubles the size of step. Buckets with odd wis perform a local upscale. Then, every bucket i updates its scale index to wi/2..

Conclusion:

A novel counters estimation data structure that minimizes the relative error. ICE-Buckets use the optimal estimation function with a scale that is optimized independently for each bucket. We first described an explicit representation of this func-tion, which was previously known only in recursive form. We extended its analysis and showed a method to measure the effect of upscale operations on the relative error. This function is used in ICE-Buckets to minimize the error in each bucket.

References

1. Sharing in MULTICS. In Proceedings of the Fourth Symposium on Operating System Principles, SOSP 1973, Thomas J. Watson, Research Center, Yorktown Heights, New York, USA, October 15-17, 1973.

2. Robert Morris and Ken Thompson. Password Security: A Case History, 1979. http://cs-www.cs.yale.edu/homes/arvind/cs422/doc/unix-sec.pdf.

3. Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In Dan Boneh, editor, Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, volume 2729 of
Lecture Notes in Computer Science, pages 617–630. Springer, 2003.

4. Password Hashing Competition (PHC), 2014. https://password-hashing.net/index.html.

5. Donghoon Chang, Arpan Jati, Sweta Mishra, and Somitra Kumar Sanadhya. Rig: A simple, secure and flexible design for password hashing. In Dongdai Lin, Moti Yung, and Jianying Zhou, editors, Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014, Revised Selected Papers, volume 8957 of Lecture Notes in Computer Science, pages 361–381. Springer, 2014.

6. Ari Juels and Ronald L. Rivest. Honeywords: making passwordcracking detectable. In 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4- 8, 2013, 2013.

7. Fred Cohen. The Use of Deception Techniques: Honeypots and Decoys. http://all.net/journal/deception/Deception Techniques .pdf.

8. Lance Spitzner. Honeytokens: The Other Honeypot, 2003. http://www.symantec.com/connect/articles/honeytokens-other-honeypot.

9. Hristo Bojinov, Elie Bursztein, Xavier Boyen, and Dan Boneh. Kamouflage: Loss-resistant password management. In Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece,

September 20-22, 2010. Proceedings, pages 286–302, 2010.

10. Wikipedia contributors. 2012 LinkedIn hack. Wikipedia, The Free Encyclopedia, Date retrieved: 29 May 2016. Available at: https://en.wikipedia.org/w/index.php?title=2012 LinkedIn hack&oldid=722095159.

11. Bruce Schneier. Cryptographic Blunders Revealed by Adobe's Password Leak. Schneier on Security, 2013. Available at: https://www.schneier.com/blog/archives/2013/11/ cryptographic b.html.

12. Swati Khandelwal. Hacking any eBay Account in just 1 minute, 2014. Available at: http://thehackernews.com/2014/09/ hacking-ebay-accounts.html.

13. Wikipedia contributors. Ashley Madison data breach. Wikipedia, The Free Encyclopedia, Date retrieved: 29 May 2016. Available at: https://en.wikipedia.org/w/index.php?title= Ashley Madison data breach&oldid=721001290.

14. Troy Hunt. Observations and thoughts on the LinkedIn data breach, 2015. Available at: https://www.troyhunt.com/ observations-and-thoughts-on-the-linkedin-data-breach/.

15. Michael Gilleland. Levenshtein Distance, in Three Flavors. Available at: http://people.cs.pitt.edu/_kirk/cs1501/assignments/editdistance/Levenshtein%20Distance.htm.