D-Sketch: A Differentiated Sketch Strategy for Double-Stack Networks

Shangsen Li, Lailong Luo, *Member, IEEE*, Deke Guo, *Senior Member, IEEE*, and Pengtao Fu

***Abstract*— Representing traffic with sketch data structures to provide flow statistics is a fundamental strategy in network measurement. In the current double-stack circumstance, we observe that IPv4 and IPv6 flows are inequivalent in terms of both flow cardinality and flow size in Internet. The existing sketches, however, represent IPv4 and IPv6 flows with no differentiation. As a consequence, the IPv4 flows, which can be massive and large, while the IPv6 flows, which are usually handful and small. Such an unbalance or asymmetry feature may lead to unnecessary hash collision errors, especially for IPv6 flows. To this end, in this letter, we present D-Sketch, a new sketch optimization strategy to represent IP flows. At its core, D-Sketch differentiates IPv4 flows from IPv6 ones, thereafter represents them with isolated sketches whose capacities are proportional to the corresponding flow cardinality. Given the same space overhead, trace-driven evaluations further commit that D-Sketch decreases the ARE of per-flow measurement up to 52.5%, with an average of 24.0%, compared with the existing one-sketch and large-small strategies.**

***Index Terms*— Network measurement, sketch, IPv6 protocol.**

I. Introduction

NETWORK traffic measurement provides basic information about the amount and type of flows for network management, security analysis, and beyond. For example, traffic engineering [13], attack and anomaly detection [19], and network planning [14], [15] must be aware of the traffic statistics. Usually, compact data structures such as CM Sketch [5], Count Sketch [4], CM-CU Sketch [8] are deployed in selected networking devices to count the number of packets in each flow.

Since the birth of IPv6, the Internet had been a double-stack network wherein both IPv4 and IPv6 are implemented. It was reported the last block of IPv4 addresses has been released and allocated. To resolve the shortage of IP addresses, IPv6 was invented and put into real use by offering 2^{128} IP addresses. Despite the consistent increasing use of IPv6, IPv4 still dominates the network. The reason for such a slow transition is many-fold, including the considerations about investment and compatibility, the adoption of emerging technologies like NAT and DHCP, and even some social reasons. Consequently, the double-stack circumstance will last for a long time.

Manuscript received October 14, 2020; accepted November 3, 2020. Date of publication November 9, 2020; date of current version March 10, 2021. This work is partially supported by the Research Plan of National University of Defense Technology under Grant No. ZK20-30, the National Natural Science Foundation of China under Grant No. 62002378, and the National Key Research under Grant No. 2018YFB1800203. The associate editor coordinating the review of this letter and approving it for publication was M. Sepulcre. *(Corresponding author: Lailong Luo.)*

The authors are with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China (e-mail: lishangsen@nudt.edu.cn; luolailong09@nudt.edu.cn; dekeguo@nudt.edu.cn; fupengtao@nudt.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2020.3036673

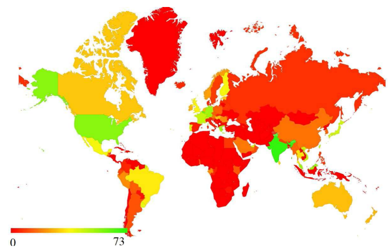


Fig. 1. IPv6 Capable Rate by country/region (%), surveyed by APNIC Labs IPv6 Measurement System [11].

Observation: **IPv4 and IPv6 flows are total inequivalent in terms of both flow cardinality and flow size in today's Internet.** For flow cardinality, the ratio between IPv6 flows and IPv4 flows is less than 1.3% in China's LTE networks [3]; Google reports that less than 30% of its users adopt IPv6 protocol [7]. Besides, as shown in Fig. 1, the IPv6 capable[1] rates of most countries remain at a low-level [11]. For flow size, as shown in Fig. 2, over 90% IPv6 flows contain less than 10 packets, while about 42% IPv4 flows include more than 10 packets. In other words, on the Internet, most large flows are propagated with IPv4 protocol.

Despite the inequivalence between IPv4 and IPv6, the existing sketches represent them with no differentiation. For instance, when CM Sketch [5] is employed, the hash functions map the incoming packets into the counter arrays directly, regardless of its IP protocol version. As a result, IPv4 and IPv6 flow coexist in the counter arrays. These flows interfere with each other due to the potential hash collisions. However, due to their domination status (in terms of both flow cardinality and flow size), IPv4 may attach non-trivial hash collisions to IPv6 flows, leading to the overestimation of IPv6 flow size. In contrast, IPv6 flows, which are usually handful and small, somehow fall short to decrease the accuracy of IPv4 flows significantly. Therefore, it is possible to improve the overall per-flow measurement accuracy of sketches by removing the interference between different types of IP flows.

Based on the above observation, in this letter, we propose D-Sketch, a differentiated sketch strategy for double-stack networks. Specifically, D-Sketch differentiates IPv4 flows from IPv6 flows and represents them with separated sketches. With such a design, D-Sketch confines the hash collisions inner a sketch and avoids interference between different types of flows. The capacity of IPv4 and IPv6 sketches is proportional to their flow cardinalities. An adaptive space allocation scheme is enabled by introducing an additional sampling mechanism into D-Sketch. To our knowledge, this is the first work, which takes the initial step to explore how to improve the accuracy

[1]IPv6 Capable means "can fetch IPv6"–The endpoint was observed to fetch a web object when the only address binding to the DNS label was an IPv6 address (IPv6 only) [10].

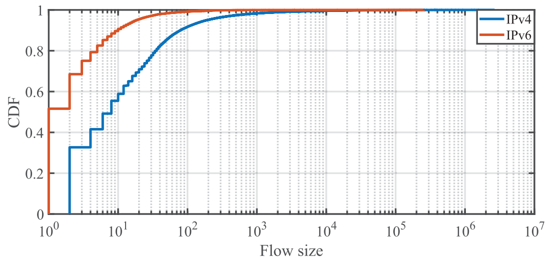


Fig. 2. The CDF of flow size in the trace collected by MAWILab. There are 180K IPv4 flows and 36K IPv6 flows, which are extracted from 82M IPv4 packets and 797K IPv6 packets in the real-world traffic trace.

of sketches in double-stack networks. The contribution of this letter can be summarized as follows.

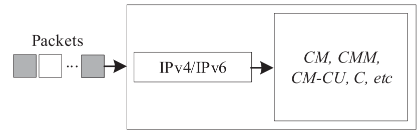
- We propose D-Sketch, a differentiated sketch strategy for Internet flows, which represents different types of IP flows with isolated sketches. D-Sketch avoids the interference between the IPv4 flows and IPv6 flows naturally.
- We present a space allocation strategy for D-Sketch, which improves the measurement accuracy with constant operation overhead and no additional space overhead.
- We implement the proposed D-Sketch strategy in diverse sketches (including CM, CMM, C, and CM-CU) and large-small strategy. Comprehensive evaluations demonstrate that the D-Sketch can effectively improve the accuracy of per-flow measurement up to 52.5%.

The remainder of this letter is organized as follows. Section II presents the related work about sketch methods. Section III introduces the framework of D-Sketch and its space allocation strategy. Section IV reports the evaluation setting and experimental results. Finally, Section V concludes this letter.

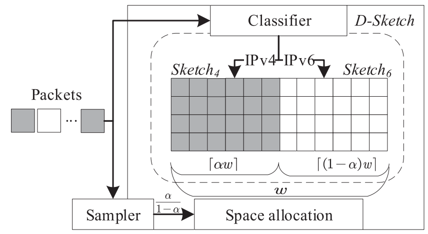
II. Related Work

CM Sketch (CM) is the most widely applied sketch for per-flow measurement [5], [6]. It consists of d arrays, $A_1 \dots A_d$, each of which contains w counters. When measuring the network traffic, each flowkey is extracted from packet header and hashed by the d independent hash functions, $h_1 \dots h_d$. Thereafter, the d corresponding counters, i.e., $\forall\ 1 \le k \le d, A_k\left[h_k\left(f\right)\%w\right]$ are updated by adding up by flow size. To query the size of a flow f, CM returns the minimum value among the d counters as the estimation.

Count Sketch (C) [4], CountMeanMin Sketch (CMM) [6] and CM-CU Sketch (CM-CU) [8] maintain the same data structure as the CM. However, they propose different updating or query strategies. C introduces one more hash function $g_k()$ for the k^{th} array to map flows onto $\{+1, -1\}$. When updating a flow f with size c, the corresponding counter will be increased by $c \cdot g_k\left(\right)$. C returns the median over d counters, i.e., $\forall\ 1 \le k \le d, median\{sk\left[k, h_k(f)\right] \cdot g_k(f)\}$, as an unbiased estimate, where $sk\left[k, h_k(f)\right]$ represents the count of flow f in the k^{th} array. CMM updates the data structure as the CM does. However, CMM formulates the noise in the k^{th} array as, $\forall\ 1 \le k \le d, \left(N - sk\left[k, h_k(f)\right]\right) / \left(w-1\right)$, where N is the cardinality of flows. After removing the noise, CMM returns the median value of d counters as the estimate. CM-CU follows the same query strategy as CM, but implements a novel update scheme. Specifically, CM-CU updates a flow f with size c as $\max\left\{sk\left[k, h_k\left(f\right)\right], \ \widehat{c}\left(f\right) + c\right\}$, where $\widehat{c}\left(f\right)$ is the query result of flow f before updating, i.e., $\forall\ 1 \le k \le d$, $\widehat{c}\left(f\right) = min\{sk\left[k, h_k(f)\right]\}$.



(a) The traditional Sketch framework



(b) The design of D-Sketch

Fig. 3. The framework of D-Sketch.

Besides, some extensive works further optimize the sketches with the joint consideration of the traffic features. Typical efforts include Cold filter [20], Elastic Sketch [18], FCM Sketch [17], SketchLearn [9] and ASketch [16], which optimize sketch with large-small strategy based on the real distribution of flow size. Another kind of sketch design, such as R-HHH [1], NitroSketch [12], and BeauCoup [2], takes randomization strategies to reduce the sketch overhead. However, we argue the existing work only consider the variety of flow size, while ignoring the inequivalence of flows with diverse IP protocols. As far as we know, this letter is the first work that tries to optimize the sketch performance in the double-stack networks. D-Sketch represents IPv4 and IPv6 independently; by contrast, existing works distinguish mouse flows from elephant flows and represent them separately. These two design philosophies are totally orthogonal.

III. The D-Sketch Design

A. The Framework of D-Sketch

As illustrated in Fig. 3(a), existing sketch-based network measurement techniques feed the employed data structures directly. By contrast, as depicted in Fig. 3(b), our D-Sketch consists of four modules, i.e., sampler, classifier, sketch, and space allocation scheme. To be specific, the sampler is responsible for determining the ratio between IPv4 flows and IPv6 flows. The classifier fetches the "version" field from the packet header to identify the IP protocol this packet involved. The sketch module divides the whole traditional sketch into two sub-sketches, labeled as sketch$_4$ and sketch$_6$, which are responsible for representing IPv4 flows and IPv6 ones, respectively. The space allocation scheme is implemented to alter the capacity of sketch$_4$ and sketch$_6$ adaptively based on the sampling result. By doing so, D-Sketch distinguishes IPv4 from IPv6 ones and represents them with isolated sketches, without extra space overhead. Accordingly, D-Sketch removes the interference in sketch between different types of flows naturally.

B. The Sampling in D-Sketch

To incorporate the characteristics of network traffic, D-Sketch runs the sampler before each measurement epoch to assist the space allocation scheme adaptively. Specifically, D-Sketch uses a very small number of packets from the traffic to estimate the IPv4/IPv6 flow ratio. To simplify the sampling procedure, we employ a fixed sampling probability for one measurement epoch. To achieve a sample rate of λ, D-Sketch saves the header of one packet after every $\lambda - 1$ packets. Obviously, a higher sampling rate leads to higher estimation accuracy. Let F represent the flow set of traffic during one measurement period and let S represent the set of flows sampled before measurement. We estimate that the IPv4/IPv6 flow ratio of F from the sample S. Because the measurement period is always about 5 to 10 minutes, it is acceptable to assume the ratio of IPv4/IPv6 maintains stable during one measurement epoch. Thus the sampling result can be applied to realize the space allocation strategy directly.

C. Space Allocation Strategy

The space allocation strategy decides the capacity of sketch$_4$ and sketch$_6$ based on the sampling result. Taking CM sketch for example, when the accuracy requirement is fixed, the setting of CM Sketch can be formulated as follows:

Theorem 1 [5]: If $w = \lceil \frac{e}{\varepsilon} \rceil$ and $d = \lceil \ln \frac{1}{\delta} \rceil$, the estimate $\hat{s}_i$ has the following guarantees: $s_i < \hat{s}_i$ with probability at least $1 - \delta$.

$$\hat{s}_i \leq s_i + \varepsilon \|f\|_1 \tag{1}$$

where s_i is the real size of the i^{th} flow in the flow set f; $\hat{s}_i$ represents the query result of the i^{th} flow from CM Sketch, and e is the base of the natural logarithm, and it is a constant chosen to optimize the space overhead for CM Sketch design.

In D-Sketch, when representing N flows with a CM sketch of parameters w and d, for any flow f, the probability that it collides with others in a d array can be expressed as:

$$p_d = \left(1 - \left(1 - \frac{1}{w}\right)^{N-1}\right)^d \approx (1 - e^{-\frac{N}{w}})^d \tag{2}$$

In D-Sketch, the value of d is given as the same. To be fair, it is advisable to remain the ratio $\frac{N}{w}$ as a constant, and allocate the w counters to IPv4 and IPv6 flows according to their cardinalities. Before a measurement epoch, D-Sketch samples the undergoing flows and thereby derives out the ratio (denote as $\frac{\alpha}{1-\alpha}$) between IPv4 flow and IPv6 flow cardinalities. Then this ratio is referred to decide the lengths of sketch$_4$ and sketch$_6$. Specifically, sketch$_4$ has $\lceil w\alpha \rceil$ counters in each array, while sketch$_6$ is assigned with$\lceil (1-\alpha)w \rceil$ counters in each array. Note that, to implement the space allocation strategy, sub-sketches need to change the range of hash functions, i.e., the location range of the d corresponding counters. Specifically, for a IPv4 flow f_4 in sketch$_4$, the counters have changed to, $\forall\ 1 \leq k \leq d, A_k\left[h_k\left(f_4\right)\%\lceil w\alpha \rceil\right]$; while for an IPv6 flow f_6 in sketch$_6$, the corresponding counters have changed to, $\forall\ 1 \leq k \leq d, A_k\left[\lceil w\alpha \rceil + h_k\left(f_6\right)\%\lceil (1-\alpha)w \rceil\right]$. Note that, D-Sketch only needs a small portion of packets to estimate the cardinality of IPv4 and IPv6 flows with a negligible space overhead. Apart from the sampling procedure, our D-Sketch incurs no more space overhead, but augments an additional operation to fetch the IP protocol field from each packet to separately represent the traffic flows into corresponding sub-sketches, which incurs a constant time complexity.

IV. Evaluation

A. Datasets

We rely on both real-world traces and synthetic datasets to evaluate our proposals. The trace was collected by MAWI Working Group from its sample point-G at 2:00 pm on 12/02/2020. This trace is further divided into 10 subsets, each of which contains about one million Internet packets. To obtain the synthetic datasets, we retype the flow size randomly. With both real-world and synthetic datasets, the experiments are sufficient to reveal the impact of flow size distribution upon D-Sketch performance. Note that all the evaluations are executed at a host with 2.6GHz CPU and 16GB RAM.

B. Parameter Setting

We evaluate the D-Sketch strategy in CM-like sketches with and without large-small strategy. Therefore, two global parameters, i.e., w and d are carefully set. Then the global sketch is divided into two parts, i.e., Sketch$_4$ for IPv4 flows and Sketch$_6$ for IPv6 flows. The length of these two parts is determined by the sampling results. And we set the threshold T for large-small flows as 15 and 255 individually.

C. Evaluation Metrics

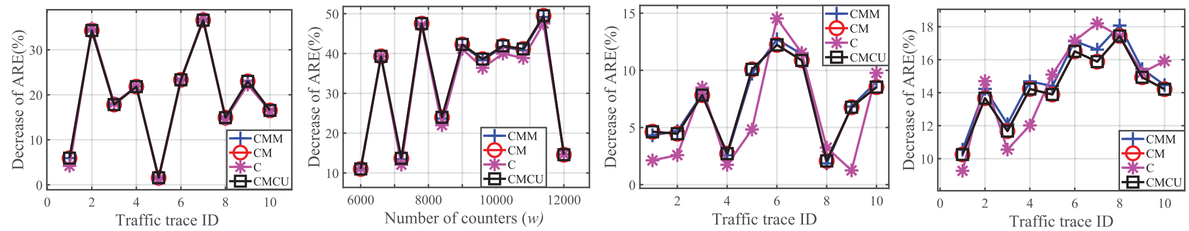
Because D-Sketch only needs a small portion of packets to estimate the cardinality of IPv4 and IPv6 flows and performs an additional operation with a constant overhead to classify different kinds of flows, we just evaluate the accuracy of D-Sketch in terms of average relative error (**ARE**), and **Decrease of ARE (DOA)** (%). Let s_i represent the actual size of the i^{th} flow in a dataset, $\hat{s}_i$ represents the estimation of flow size, and N counts the total number of flows. Then ARE is defined as $\frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{s}_i - s_i|}{s_i}$, and the Decrease of ARE is defined as the ratio between the ARE achieved by the D-Sketch and the ARE achieved by the sketch without the differentiation strategy, i.e., $\frac{ARE - ARE_D}{ARE} \times 100\%$. The metric **ARE** exactly depicts the performance when applying sketches on per-flow measurement. The metric **DOA** explicitly quantifies the improvement of accuracy on per-flow measurement when D-Sketch strategy is implemented.

D. Evaluation Results

We evaluate the performance when applying the D-Sketch strategy on different sketches, including CM, CMM, CM-CU, and C, with/without the large-small strategy. The sketch capacities are specified according to flow cardinality. The value of w is set as 6000 and d is set as 1.

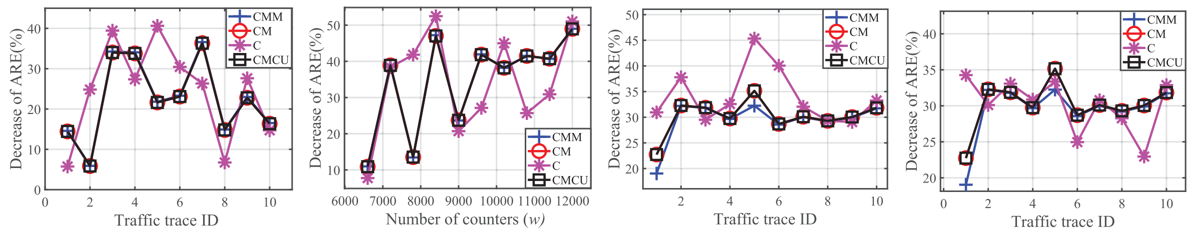
As shown in Fig. 4, the ARE of all the sketches with D-Sketch decrease significantly. For real-world traffic traces, when applying D-Sketch on sketches without the large-small strategy, the DOA ranges from 1.2% to 37.0% with an average of 19.5%. When changing w, the DOA lies in the range $[11.0\%, 50.0\%]$ with the average of 32.7%. When applying D-Sketch on the large-small strategy, the DOA ranges in [9.3%,18.2%] with an average of 14.5% when $T = 255$, and [1.2%,14.5%] with an average of 6.8% when $T = 15$.

As illustrated in Fig. 5, for synthetic datasets, when applying D-Sketch on sketches without the large-small strategy,



(a) For different traffic trace. (b) When w varies. (c) $T = 15$ in large-small strategy (d) $T = 255$ in large-small strategy

Fig. 4. When applying D-Sketch strategy on real traffic trace. The first (last) two subgraphs quantify the sketches without (with) large-small strategy.



(a) For different traffic trace. (b) When w varies. (c) $T = 15$ in large-small strategy (d) $T = 255$ in large-small strategy

Fig. 5. When applying D-Sketch strategy on synthetic datasets. The first (last) two subgraphs quantify the sketches without (with) large-small strategy.

the DOA ranges in [5.9%, 40.7%] with an average of 22.9%. When changing w, the DOA lies in the range $[7.7\%, 52.5\%]$ with an average of 34.4%. When applying D-Sketch in large-small strategy, the DOA ranges in [19.0%, 35.1%] with an average of 30.0% when $T = 255$, and [19.0%,35.1%] with an average of 31.0% when $T = 15$.

The above evaluations conclude that the implementation of D-Sketch in existing sketches helps improve the accuracy of Internet measurement. The fundamental reason is that, by isolating the two kinds of IP flows and representing them with independent sketches, the interference between them is removed naturally. Essentially, the impact of inequivalence between IPv4 and IPv6 is well controlled by D-Sketch strategy.

V. Conclusion

In this letter, we observe IPv4 and IPv6 flows are inequivalent in terms of both flow cardinality and flow size in today's Internet. To avoid the interference between these two types of flows, we propose D-Sketch, a differentiated sketch strategy for double-stack networks. We further design a space allocation strategy for D-Sketch, which assign the space to IPv4 and IPv6 flows according to their sampled flow cardinality. Comprehensive experiments demonstrate D-Sketch can efficiently improve the accuracy of per-flow measurement.

References

[1] R. Ben Basat, G. Einziger, R. Friedman, M. C. Luizelli, and E. Waisbard, "Constant time updates in hierarchical heavy hitters," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Los Angeles, CA, USA, Aug. 2017, pp. 127–140.

[2] X. Chen, S. L. Feibish, M. Braverman, and J. Rexford, "BeauCoup: Answering many network traffic queries, one memory update at a time," in *Proc. SIGCOMM*, New York, NY, USA, Aug. 2020, pp. 226–239.

[3] Coogle. (May 2020). *Ipv6–Google*. Accessed: May 26, 2020. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html

[4] G. Cormode and M. Hadjieleftheriou, "Finding frequent items in data streams," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1530–1541, Aug. 2008.

[5] G. Cormode and S. Muthukrishnan, "An improved data stream summary: The count-min sketch and its applications," in *Proc. LATIN*, Buenos Aires, Argentina, Apr. 2004, pp. 29–38.

[6] F. Deng and D. Rafiei. *New Estimation Algorithms for Streaming Data: Count-Min Can do More*. Accessed: Oct. 20, 2020. [Online]. Available: http://webdocs.cs.ualberta.ca/~drafiei/papers/cmm.pdf

[7] Committee for Advancing the Extensive Deployment of IPv6, "The develop of IPv6 in China," in *Proc. China Internet Conf.*, Beijing, China, Jul. 2019.

[8] A. Goyal, H. D. III, and G. Cormode, "Sketch algorithms for estimating point queries in NLP," in *Proc. MNLP-CoNLL*, Jeju Island, South Korea, Jul. 2012, pp. 1093–1103.

[9] Q. Huang, P. P. C. Lee, and Y. Bao, "Sketchlearn: Relieving user burdens in approximate measurement with automated statistical inference," in *Proc. SIGCOMM*, Budapest, Hungary, Aug. 2018, pp. 576–590.

[10] G. Huston, J. Damas, and G. Michalson, "Measuring IPv6 using ad-based measurement," APNIC Labs, Brisbane, QLD, Australia. Accessed: Apr. 22, 2020. [Online]. Available: https://labs.apnic.net/

[11] S. Lab. (Apr. 2020). *IPv6 Measurement Maps*. Accessed: Apr. 22, 2020. [Online]. Available: https://stats.labs.apnic.net/ipv6/

[12] Z. Liu *et al.*, "Nitrosketch: Robust and general sketch-based monitoring in software switches," in *Proc. SIGCOMM*, Beijing, China, Aug. 2019, pp. 334–350.

[13] L. Luo, D. Guo, R. T. B. Ma, O. Rottenstreich, and X. Luo, "Optimizing Bloom filter: Challenges, solutions, and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1912–1949, 2nd Quart., 2019.

[14] L. Luo, D. Guo, O. Rottenstreich, R. T. B. Ma, X. Luo, and B. Ren, "The consistent cuckoo filter," in *Proc. IEEE IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 712–720.

[15] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, "Trumpet: Timely and precise triggers in data centers," in *Proc. SIGCOMM*, Florianopolis, Brazil, Aug. 2016, pp. 129–143.

[16] P. Roy, A. Khan, and G. Alonso, "Augmented sketch: Faster and more accurate stream processing," in *Proc. SIGMOD*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 1449–1463.

[17] D. Thomas, R. Bordawekar, C. C. Aggarwal, and P. S. Yu, "On efficient query processing of stream counts on the cell processor," in *Proc. IEEE 25th Int. Conf. Data Eng.*, Shanghai, China, Mar. 2009, pp. 748–759.

[18] T. Yang *et al.*, "Elastic sketch: Adaptive and fast network-wide measurements," in *Proc. SIGCOMM*, Budapest, Hungary, Aug. 2018, pp. 561–575.

[19] W. Zhang, F. Kong, L. Yang, Y. Chen, and M. Zhang, "Hierarchical community detection based on partial matrix convergence using random walksk," *Tsinghua Sci. Technol.*, vol. 23, no. 1, pp. 35–46, Feb. 2018.

[20] Y. Zhou *et al.*, "Cold filter: A meta-framework for faster and more accurate stream processing," in *Proc. SIGMOD*, Houston, TX, USA, Jun. 2018, pp. 741–756.