

数据流测量的非对称主备切换机制

张 进, 黄清杉, 赵文栋, 彭来献

(解放军理工大学通信工程学院, 南京 210007)

摘 要: 针对现有的数据流测量算法所采用的对称主备切换机制存在存储空间浪费的问题, 提出一种非对称主备切换机制, 用于提高数据流测量算法的空间效率。非对称主备切换机制利用网络数据流达到过程的平稳特性, 以及 DRAM 的高速块写入特性, 只需采用一个小容量的备存储器, 即可实现主备切换的功能。实验结果表明, 与对称主备切换机制相比, 非对称主备切换机制在几乎不影响测量错误概率的前提下, 能够节省近一半的存储器空间。

关键词: 网络流量分析; 网络流量测量; 布鲁姆过滤器; 主备切换; 错误概率; 空间效率

Asymmetrical Primary-standby Switching Mechanism for Data Flow Measurement

ZHANG Jin, HUANG Qing-shan, ZHAO Wen-dong, PENG Lai-xian

(Institute of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China)

【Abstract】 The Symmetrical Primary Standby Switching(SPSS) mechanism employed by existing flow traffic measurement algorithms can not make full use of memory space. An Asymmetrical Primary Standby Switching(APSS) mechanism is presented for improving the space efficiency of data flow measurement algorithms. The APSS mechanism is based on the observations that flow arrival process is stable, and DRAM can support bulk write which is much faster than random access. A small standby memory is enough to realize the primary-standby mechanism. Experimental results show that compared with SPSS, APSS can reduce memory consumption by almost a half while having trivial impact on the measurement error probability.

【Key words】 network traffic analysis; network traffic measurement; Bloom filter; primary-standby switching; error probability; space efficiency

DOI: 10.3969/j.issn.1000-3428.2013.12.054

1 概述

网络流量测量对于计费管理、业务流量控制、网络异常检测以及网络安全监测等具有重要的意义^[1-2]。在高速 IP 网络中, 由于数据包到达间隔短, 且并发数据流的数目巨大, 若直接进行流量测量, 实现代价极高。现有的骨干网流量测量算法从不同的角度降低骨干网流量测量的代价, 主要包括抽样法、长流法和近似法。抽样法随机抽取一些数据包, 根据抽样结果, 对原始数据流的流量进行估计。当前, Cisco 的 Netflow 所采用的就是抽样法^[3]。对于流量较大的数据流, 抽样法可以获得较为准确的测量结果; 而对于流量较小的数据流, 抽样法往往导致较大的测量误差, 甚至漏测了许多流量较小的数据流。根据对网络流量进行采集分析的结果, 网络数据流的流量大小服从 Zipf 分布,

约 20% 的长流, 其流量之和占据链路总流量的 80%^[4-5]。长流法试图对流量较大的长流进行准确的测量, 而忽略流量小的短流^[5-6]。可见, 长流法和抽样法均侧重于对长流的测量, 然而, 对于网络异常检测、网络业务流分类而言, 忽略短流会遗漏大量重要的信息。与抽样法和长流法不同, 近似法试图对所有的数据流的流量进行测量, 但是测量结果存在一定的误差。近似法又分为 2 类, 第 1 类算法首先对流量值进行在线压缩, 然后通过离线的解压过程, 估算各条数据流的流量^[7-8]。此类算法无法支持流量值的在线查询, 称这类算法为被动式近似测量算法^[9-11]。第 2 类近似测量算法采用计数型布鲁姆过滤器实现, 主要包括 CBF^[12]、dlCBF^[13]和 BSdlCBF^[14], 可以支持流量值的在线查询, 称这类算法为主动式近似测量算法。显然, 主动式近似测量算法具有最广泛的应用范畴。

基金项目: 江苏省自然科学基金资助项目(BK2010103)

作者简介: 张 进(1979—), 男, 工程师、博士, 主研方向: 网络测量, 网络编码; 黄清杉, 助理工程师、硕士; 赵文栋、彭来献, 副教授、博士

收稿日期: 2012-08-17 **修回日期:** 2012-11-13 **E-mail:** boost_zj@163.com

近似测量算法的性能主要受到高速存储器资源的制约。在待测数据流的数目一定的前提下, 可用的高速存储器空间越大, 测量的精度越高。现有的近似测量算法所采用的对称的主备切换机制对于高速存储器的空间利用效率不高。本文提出一种非对称的主备切换机制, 用于提高近似测量算法的空间效率。

2 对称与非对称主备切换机制

在流测量过程中, 流状态、流量计数器需要保存在高速存储器中, 以便实现逐包查询与更新。由于高速存储器的空间极为有限, 因此, 当某条数据流结束时, 应当将其所有信息从高速存储器中清除, 以便为后继到达的数据流释放存储资源。但是, 现有的数据流流量的近似测量算法不能支持已结束的数据流释放其所占用的资源, 其原因主要有以下 2 点:

(1) 某些流测量算法在内在机制设计上就不能够支持删除操作, 例如 SCBF^[7]、BSdlCBF^[14]、CB^[8-9]。

(2) 另外一些流测量算法虽然能够支持数据流的删除操作, 如 CBF^[12]、dlCBF^[13], 但是, 由于哈希冲突, 会导致漏删除的现象——某些数据流虽然已经结束, 但其流量计数器仍然在增长, 因而无法将其删除。随着漏删数据流的累积, 也会导致系统资源耗尽。

因此, 现有的流测量算法通常采用 2 个独立的高速存储器, 通过主备切换的方式, 交替进行工作。以 SCBF 为例说明现有的流测量算法中所应用的主备切换机制。图 1 给出了 SCBF 的结构模型。SCBF 采用了 2 块相同容量的高速存储器 SRAM1 和 SRAM2, 假设在第 i 个测量周期, SRAM1 用于更新, SRAM2 用于查询, 则在第 $i+1$ 个测量周期, SRAM2 用于更新, SRAM1 用于查询。查询完成后, 需要将查询的 SRAM 清零, 以便在下一个测量周期将其用于更新^[14]。

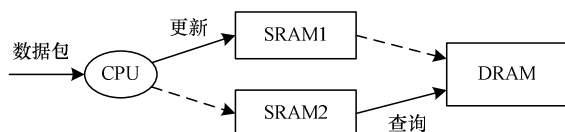


图 1 SCBF 所采用的主备切换结构

主备切换机制虽然解决了对高速存储器中已结束数据流的删除问题, 但是, 主备切换机制使得流测量算法对于高速存储器的容量需求倍增。此外, 备存储器在整个测量周期中, 仅需要支持对上一周期测量结果的查询, 被访问的次数相对主存储器而言极为有限, 其带宽并没有得到充分的利用。可见, 主备切换机制在高速存储器的有效利用方面存在很大的缺陷。针对现有的流测量算法中主备切换机制的缺陷, 提出一种非对称的主备切换机制, 以提高流测量算法中高速存储器的利用效率。非对称主备切换机制的提出主要基于下列事实:

与高速存储器相比, 低速存储器的容量较大。因此,

可以在低速存储器中为高速存储器建立镜像, 在测量周期结束时, 立即将高速存储器中的数据批量复制到低速存储器中, 之后, 所有的查询操作仅需要访问低速存储器即可。虽然低速存储器(通常是 DRAM)的随机访问带宽较低, 但是, DRAM 可以支持块写入(Bulk Write)方式, 其等效带宽可以比随机访问时高出数倍。因此, 完成主存储器的镜像复制所需的时间极短。例如, 假设 DRAM 块写入的带宽为 1 Gb/s, 高速存储器的容量为 100 MB, 则完成镜像复制所需的时间仅为 10 ms。测量周期的长度通常为数分钟, 对比而言, 镜像复制时间极短。

现有的网络测量的研究结果表明, 数据流的到达过程较为平稳, 到达时间间隔基本服从 Poisson 分布^[4, 15]。因此, 在镜像复制时间(数十毫秒)内所到达的数据流仅占整个测量周期(数分钟)内所有数据流的极小比例。可以采用一个小容量的备存储器, 用于测量在镜像复制时间内到达的数据流; 在镜像复制时间结束后, 被复制的高速存储器可以接替上来, 继续用于当前周期的测量。

基于上述思路, 提出了流测量的非对称主备切换机制, 其基本结构如图 2 所示。非对称主备切换机制采用 3 块独立的存储器, 分别是主存储器、备存储器和活动存储器, 同时, 在非对称主备切换机制中, 测量周期划分为镜像建立时间和非镜像建立时间 2 个部分。不妨假设在第 i 个测量周期的镜像建立时间内, 主存储器和活动存储器用于查询(向低速存储器中复制), 备存储器用于更新; 在第 i 个测量周期的非镜像建立时间内, 主存储器空闲, 备存储器和活动存储器用于更新。则在第 $i+1$ 个测量周期的镜像建立时间内, 备存储器和活动存储器用于查询(向低速存储器中复制), 主存储器用于更新; 在第 i 个测量周期的非镜像建立时间内, 备存储器空闲, 主存储器和活动存储器用于更新。可见, 在非对称主备切换机制中, 在任意时间, 最多有主存储器或者备存储器中的某一个处于空闲状态; 作为对比, 在现有的对称主备切换机制中, 始终有一半存储器基本处于空闲状态。主/备存储器的空间相对于活动存储器的空间而言较小, 因此, 与对称主备切换相比, 非对称主备切换机制大约可以节省一半的高速存储器空间。

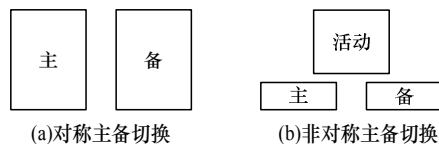


图 2 对称主备切换和非对称主备切换的比较

3 基于非对称主备切换的 CBF 流测量算法

计数型布鲁姆过滤器(Counting Bloom Filter, CBF)由 k 个独立的哈希函数 $h_1(\cdot), h_2(\cdot), \dots, h_k(\cdot)$ 和一个计数器向量 C 构成^[12], 如图 3 所示。设 C 中共包含了 m 个计数器, 每个计数器的宽度为 w 比特。每到达一个数据包, 取得其流标识 f , 然后将计数器 $C[h_1(f)], C[h_2(f)], \dots, C[h_k(f)]$

分别增加 1。查询 f 的流量 c_f 时, 取计数器 $C[h_1(f)]$, $C[h_2(f)], \dots, C[h_k(f)]$ 中最小的作为 c_f 的估计值, 即:

$$\hat{c}_f = \arg \min_i (C[h_i(f)]), i \in [1, k]$$

CBF 的测量错误概率为^[12]:

$$E_C = \left(1 - \left(1 - \frac{1}{m}\right)^{kN}\right)^k \approx \left(\frac{1}{2}\right)^k$$

其中, N 为测量周期内数据流的总数, $k = \ln(2) \cdot m / N$ 。

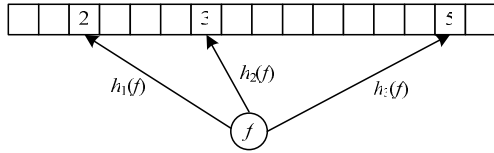


图3 CBF 的结构示意图

在 CBF 中应用非对称主备切换机制时, 需要将计数器向量的存储方式按照图 4 的方式进行组织。每个计数器划分为 A 、 B 、 C 3 个区域, A 区域作为活动存储区, B 区域和 C 区域作为主备存储区。 A 区域的位宽为 $w-rw$ 比特, B 区域和 C 区域的位宽均为 rw 比特, 计数值的高位存放在 A 区域中, 低位存放在 B 区域或者 C 区域中。称 r ($0 < r < 1$) 为备份系数, 其含义是备份存储区的大小占主存储区和活动存储区大小之和的比例。对于传统的对称主备切换机制而言, $r=1$ 。

在 CBF 中, 非对称主备切换机制的工作方式如下: 假设 t_0, t_1, t_2 为连续的 3 个测量周期的起始时刻, 镜像建立时间为 ΔT 。假设在 t_0 时刻, A 、 B 用于查询(即向低速存储器中复制镜像), C 用于更新; 在 $t_0 + \Delta T$ 时刻, A 和 B 中的内容已经复制到低速存储器中, 从这一时刻开始, A 、 C 用于更新, B 空闲。在 t_1 时刻, A 、 C 用于查询, B 用于更新; 在 $t_1 + \Delta T$ 时刻, A 、 B 用于更新, C 空闲。按照上述过程, 在相邻的测量周期循环往复, 如图 4 所示。

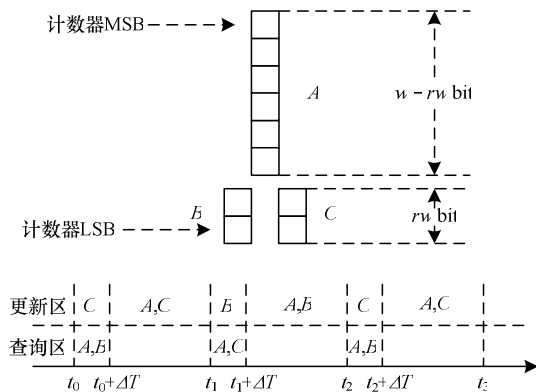


图4 面向 CBF 的非对称主备切换机制

与基于对称主备切换的 CBF 相比, 基于非对称主备切换的 CBF 在镜像建立时间 ΔT 内, 其有效计数器位宽为 rw 比特, 而非 w 比特。因此, 如果某些计数器在镜像建立时间内溢出, 将会导致额外的测量误差。然而, 由于镜像建

立时间相比测量周期而言非常短, 并且考虑到网络数据流流量服从 Zipf 分布, 流量大的流仅占极少的部分^[4-6], 因此, 当 r 足够大时, 由于镜像建立时间内计数器溢出而导致的误差基本可以忽略。

4 基于非对称主备切换的 dlCBF 流测量算法

dlCBF 将存储区划分为 d 个等长的块, 每块包含若干个桶, 每个桶划分为若干个单元, 一个桶单元存放一条数据流的流指纹和流量计数器^[13]。图 5 给出 dlCBF 的结构示意图。dlCBF 的存储区划分为 $d=4$ 个块, 分别是 BV_1 、 BV_2 、 BV_3 和 BV_4 , 每块存储区中包含 5 个桶, 每个桶中包含 4 个单元, 桶单元用于存放数据流的流指纹和流量计数器。

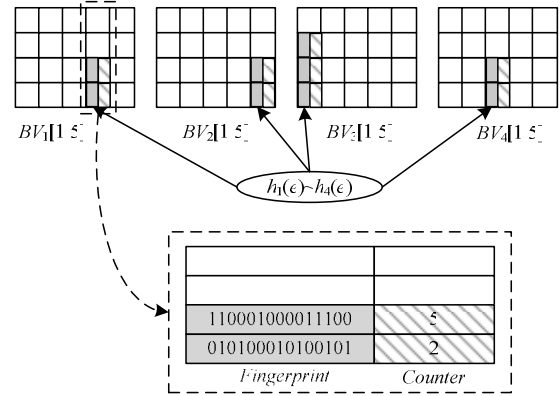


图5 dlCBF 的结构示意图

dlCBF 更新操作的步骤如下: 首先, 根据数据流的流标识, 不妨设为 f , 利用一个哈希函数 $H(\cdot)$, 计算该数据流的流指纹, 假设为 p , 即 $p = H(f)$; 然后, 依次在桶 $BV_1[h_1(f)]$, $BV_2[h_2(f)], \dots, BV_d[h_d(f)]$ 中进行查找, 若某个桶中存放有和 p 一致的流指纹, 则将对应该桶单元的流量计数器增加 1, 其中, $h_1(\cdot), h_2(\cdot), \dots, h_d(\cdot)$ 为 d 个哈希函数, 用于计算数据流 f 在各个块中的桶地址; 若 $BV_1[h_1(f)]$, $BV_2[h_2(f)], \dots, BV_d[h_d(f)]$ 中均没有和 p 一致的流指纹, 则表明数据流 f 是首次更新。此时, 令其流量计数器值为 1, 并将其流指纹 p 和流量计数器放入到 $BV_1[h_1(f)]$, $BV_2[h_2(f)], \dots, BV_d[h_d(f)]$ 这 4 个桶中负载最轻的那个桶中去; 若负载最轻的桶有多个, 则选择最左边那个桶。

dlCBF 查询操作的步骤如下: 首先, 根据流标识 f , 计算得到流指纹 p ; 然后, 依次在桶 $BV_1[h_1(f)]$, $BV_2[h_2(f)], \dots, BV_d[h_d(f)]$ 中进行查找, 将这 d 个桶中所有流指纹等于 p 的桶单元中的流量计数器累加起来, 作为待查询的数据流的流量的估计值。由于不同的流标识经过哈希运算可能产生相同的流指纹, 因此查询结果可能会产生错误。假设 dlCBF 共由 d 个块组成, 平均桶负载为 b , 流指纹长度为 p 比特, 则 dlCBF 的错误概率为:

$$E_d = 1 - \left(1 - \frac{1}{2^p}\right)^{d \cdot b} \approx \frac{d \cdot b}{2^p}$$

面向 dICBF 的非对称主备切换机制如图 6 所示。dICBF 的每个桶划分为 A 、 B 和 C 3 个区域, A 为活动区域, B 和 C 为主备切换区域。 A 区域包含 $b-rb$ 个桶单元, B 和 C 区域均包含 rb 个桶单元。在镜像建立时间内, B 或者 C 中某一个用于更新, 另外一个和 A 一起用于查询; 在非镜像建立时间内, B 或者 C 中某一个和 A 一起用于更新, 另外一个处于空闲状态。各个存储区域的工作顺序和 CBF 中的非对称主备切换机制类似, 这里不再赘述。

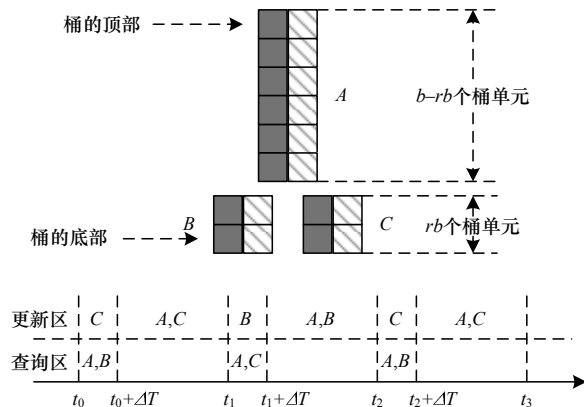


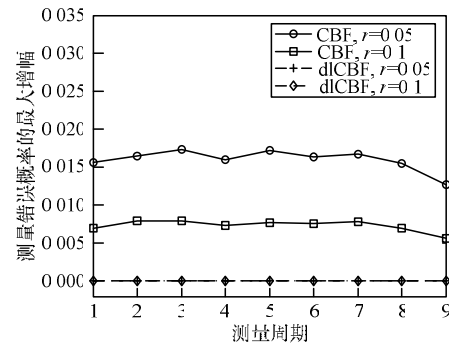
图 6 面向 dICBF 的非对称主备切换机制

与采用对称主备切换机制的 dICBF 相比, 基于非对称主备切换机制的 dICBF 可能因为在镜像建立时间内, 桶深仅为 rb , 而非 b 。因此, 基于非对称主备切换机制的 dICBF 有可能因为在镜像建立时间内达到的数据流的数目超出了设计预期而引起额外的误差。然而, 考虑到网络数据流到达过程的平稳特性^[15], 以及镜像建立时间在整个测量周期中所占的极小比例, 只要 r 给予一定的设计冗余, 基本可以避免这部分误差。

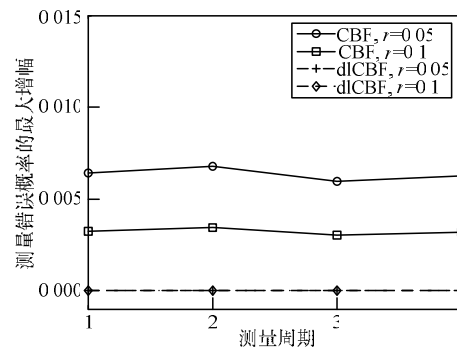
5 仿真实验

本节采用真实的骨干网流量数据, 对基于非对称主备切换的 CBF 和 dICBF 的性能进行仿真分析。实验所采用的流量数据来源于 CAIDA 所提供的 Abilene-I 和 Abilene-III 骨干网流量数据, 前者采集自 OC-48 链路, 后者采集自 OC-192 链路。分别对 Abilene-I 中的 IPLS-KSCY-20020814-90000-0.gz 流量数据文件和 Abilene-III 中的 20040601-193121-0.gz 流量数据文件的前 2 GB 进行了处理, 将其由 ERF(Extended Record Format)格式转化为本文实验所需要的数据流级的流量数据, 下文将这 2 条流量数据分别称作 OC-48 Trace 和 OC-192 Trace。在转换过程中, 采用<源 IP, 目的 IP, 源端口号, 目的端口号, 协议类型>五元组作为流标识, 测量周期的长度为 1 min。OC-48 Trace 的前 2 GB 数据时长共 583.778 s, 包含 9 个完整的测量周期; OC-192 Trace 的前 2 GB 数据时长共 264.745 s, 包括 4 个完整的测量周期。假设高速存储器的镜像大小为 100 MB, 低速存储器块写入的速度为 1 Gb/s, 则镜像建立时间为 0.1 s。备份系数 r 分别取 0.05 和 0.1。

图 7 给出了基于非对称主备切换的 CBF 和 dICBF 的测量错误概率相对于对称主备切换的 CBF 和 dICBF 的测量错误概率的最大增幅。



(a) OC-48 Trace



(b) OC-192 Trace

图 7 CBF 和 dICBF 测量错误概率的最大增幅

由图 7 可见, 基于非对称主备切换的 CBF 相对于对称主备切换的 CBF 而言, 其测量错误概率略有增加。这是因为 r 较小时, 在镜像建立时间内, CBF 的计数器位宽较窄, 导致部分数据流的流量计数器溢出, 因而导致测量错误概率略有增加。需要指出的是, 在实验中, 对于 CBF 而言, 认为所有在镜像建立时间内到达的数据流均会发生测量错误。这一假设是比较保守的, 因为网络数据流的流量服从 Zipf 分布^[4-5], 大部分数据流的流量较小, 即使在镜像建立时间内计数器的位宽较窄, 也有部分数据流的流量计数器不会发生溢出, 从而可能不会发生测量错误。因此, 实验给出的结果是测量错误概率的最大增幅。对于 dICBF 而言, 采用非对称主备切换机制不会导致测量错误的增加。这是因为尽管 r 取值较小, 例如 $r=0.05$, 但镜像建立时间内数据流的数目和测量周期内总的的数据流数目的比值比 r 更小。例如, 对 OC-192 Trace 而言, 第 1 个~第 4 个测量周期内数据流的总数目依次为 172 004 条、175 985 条、179 514 条和 180 604 条, 而第 1 个~第 4 个测量周期的镜像建立时间所到达的数据流的数目依次为 2 719 条、3 048 条、2 664 条和 2 879 条。可见, 基于非对称主备切换的 dICBF 即使当 r 较小时, 主/备存储区也完全能够容纳在镜像建立时间内所到达的数据流, 不会发生溢出现象。由于上述原因, 基于非对称主备切换的 dICBF 和对称主备切换的 dICBF 的测量错误概率才会相等。

6 结束语

本文提出了一种非对称主备切换机制, 用于提升流测量算法的空间效率。分别基于 CBF 和 dlCBF, 给出了非对称主备切换机制的具体实施方式。采用真实骨干网流量数据的仿真结果表明, 该机制可以节省约一半的高速存储器空间, 且几乎不影响测量错误概率。

本文所提出的非对称主备切换机制同样可以应用到其他流测量算法中, 如 BSdlCBF^[14]、CB^[8-9]。下一步的研究工作将考虑非对称主备切换机制在上述算法中的具体应用, 并评估应用效果。

参考文献

- [1] Trammell B, Boschi E. An Introduction to IP Flow Information Export[J]. IEEE Communications Magazine, 2011, 49(4): 89-95.
- [2] IETF. IP Flow Information Export(IPFIX) Working Group[EB/OL]. (2001-09-27). <http://www.ietf.org/html.charters/ipfix-charter.html>.
- [3] Cisco Systems Inc.. NetFlow[EB/OL]. (2004-10-31). <http://www.cisco.com/web/go/netflow>.
- [4] Wang Fang, Peterson L. Inter-AS Traffic Patterns and Their Implications[C]//Proc. of Global Telecommunications Conference. [S. l.]: IEEE Press, 1999: 1859-1868.
- [5] Estan C, Varghese G. New Directions in Traffic Measurement and Accounting[C]//Proc. of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York, USA: ACM Press, 2002: 323-336.
- [6] 吴桦, 龚俭, 杨望. 一种基于双重 Counter Bloom Filter 的长流识别算法[J]. 软件学报, 2010, 21(5): 1115-1126.
- [7] Kumar A, Xu Jun, Wang Jia. Space-code Bloom Filter for Efficient Per-flow Traffic Measurement[J]. IEEE Journal on Selected Areas in Communications, 2004, 24(12): 2327-2339.
- [8] Lu Yi, Montanari A. Counter Braids: A Novel Counter Architecture for Per-flow Measurement[C]//Proc. of International Conference on Measurement and Modeling of Computer Systems. New York, USA: ACM Press, 2008: 121-132.
- [9] Lu Yi, Prabhakar B. Robust Counting Via Counter Braids: An Error-resilient Network Measurement Architecture[C]//Proc. of IEEE INFOCOM2009. [S. l.]: IEEE Press, 2009: 522-530.
- [10] Lieven P, Scheuermann B. High-speed Per-flow Traffic Measurement with Probabilistic Multiplicity Counting[C]//Proc. of IEEE INFOCOM2010. [S. l.]: IEEE Press, 2010: 1-9.
- [11] Li Tao, Chen Shigang, Ling Yibei. Fast, Compact Per-flow Traffic Measurement Through Randomized Counter Sharing[J]. IEEE Transactions on Networking, 2011, 20(5): 1622-1634.
- [12] Li Fan, Peng Cao, Almeida J, et al. Summary Cache: A Scalable Wide-area Web Cache Sharing Protocol[J]. IEEE/ACM Transactions on Networking, 2000, 8(3): 281-293.
- [13] Bonomi F, Mitzenmacher M, Panigrahy R, et al. An Improved Construction for Counting Bloom Filters[C]//Proc. of European Symposium on Algorithms. Sydney, Australia: [s. n.], 2006: 678-680.
- [14] 张进, 邬江兴, 刘勤让. 4种计数型 Bloom Filter 的性能分析与比较[J]. 软件学报, 2010, 21(5): 1098-1114.
- [15] 周明中. 大规模网络 IP 流行为特性及其测量算法研究[D]. 南京: 东南大学, 2006.

编辑 顾逸斐

(上接第 250 页)

参考文献

- [1] Aggrawal C, Han Jiawei, Wang J, et al. A Framework for Clustering Evolving Data Streams[C]//Proc. of the 29th VLDB Conference. Berlin, Germany: IEEE Computer Society, 2003.
- [2] Cao Feng, Ester M, Qian Weining, et al. Density-based Clustering over an Evolving Data Stream with Noise[C]//Proc. of SIAM International Conference on Data Mining. Bethesda, USA: Springer, 2006.
- [3] Chen Yixin. Density-based Clustering for Real-time Stream Data[C]//Proc. of International Conference on Knowledge Discovery and Data Mining. Sacramento, USA: [s. n.], 2007.
- [4] 胡睿, 林昭文, 柯宏力, 等. 一种基于密度和滑动窗口的数据流聚类算法[J]. 计算机科学, 2011, 38(5): 145-148.
- [5] 章季阳, 王伦文. 一种领域覆盖的数据流聚类算法[J]. 小型微型计算机系统, 2012, 33(9): 1913-1916.
- [6] 曹锋, 周傲英. 基于图形处理器的数据流快速聚类[J]. 软件学报, 2007, 30(2): 291-302.
- [7] Ruiz C, Menasalvas E, Spiliopoulou M. C-DenStream: Using Domain Knowledge on a Data Stream[C]//Proc. of the 12th International Conference on Discovery Science. Porto, Portugal: Springer-Verlag, 2009.
- [8] Antonellis P, Makris C, Tsirakis N. Algorithms for Clustering Clickstream Data[J]. Information Processing Letters, 2009, 109(8): 381-385.
- [9] 杨宁, 唐常杰, 王悦, 等. 一种基于时态密度的倾斜分布数据流聚类算法[J]. 软件学报, 2010, 21(5): 1031-1041.
- [10] Kranen P, Assent I. Self-adaptive Anytime Stream Clustering[C]//Proc. of the 9th IEEE International Conference on Data Mining. [S. l.]: IEEE Computer Society, 2009.
- [11] Kranen P, Reidl F, Villaamil F, et al. Hierarchical Clustering for Real-time Stream Data with Noise[C]//Proc. of the 23rd International Conference on Scientific and Statistical Database Management. Portland, USA: Springer-Verlag, 2011.
- [12] 邵峰晶, 于忠清. 数据挖掘原理与算法[M]. 北京: 中国水利水电出版社, 2009.
- [13] Tan Pangning, Michael S, Vipin K. 数据挖掘导论[M]. 2版. 范明, 范宏建, 译. 北京: 人民邮电出版社, 2011.

编辑 索书志