中国科学: 信息科学 2024 年 第 54 卷 第 7 期: 1677–1691
SCIENTIA SINICA Informationis
论文

基于自适应 Sketch 的高速网络流大小测量机制

卜霄菲[1], 黄河[2*], 孙玉娥[3], 王兆杰[2], 吴晓灿[2]

1. 沈阳师范大学软件学院, 沈阳 110034
2. 苏州大学计算机科学与技术学院, 苏州 215006
3. 苏州大学轨道交通学院, 苏州 215131
* 通信作者. E-mail: huangh@suda.edu.cn

收稿日期: 2023–10–08; 修回日期: 2024–01–24; 接受日期: 2024–03–21; 网络出版日期: 2024–07–09

国家自然科学基金 (批准号: 62332013, 62072322, 62202322, U20A20182) 项目资助

摘要 高速网络流大小的测量面临着高速存储资源极度稀缺的挑战, 难以满足海量流式数据的实时存储需求. 目前的研究大多采用存储资源共享技术, 以便将设计的估计器置于稀缺的高速片上缓存中. 然而, 这种方法引入了大量难以消除的噪声, 导致中小规模流的估算精度不高. 为了解决这一问题, 本文提出一种能根据流大小自适应调整所占用存储空间的自适应 Sketch 技术, 并在此基础上设计出一个高精度、低存储开销的每流大小估计器. 自适应 Sketch 技术利用可逆计数器高效滤除海量噪声小流, 并进一步采用采样概率逐层递减的采样计数器实现对不同规模流的自适应采样计数, 从而控制大流对资源的过多占用, 实现了低开销、高精度的每流大小测量. 基于真实网络数据集 CAIDA 2019 的仿真实验表明, 所提出的自适应 Sketch 流大小估计器的平均相对误差较现有机制降低了接近 1 个数量级.

关键词 高速网络, 流大小测量, Sketch, 噪声小流过滤, 可逆计数器

1 引言

高速网络实时流量测量具有广泛的应用, 可以为诸如路由决策、资源调度、安全等网络重要模块或应用提供数据支持. 在实际应用中, 需要测量的网络流可以是 TCP 流、P2P 流、Http 流或者是根据应用需求抽象出共享同一流标签的所有数据包组成的流. 例如, 可以将所有发送到同一目的地址的所有数据包抽象为一条目的地址流, 其中目的地址可以抽象为该流的流标签, 而该流中所包含的数据包数量称为该流的大小. 近年来, 随着工业互联网、物联网、边缘计算以及流媒体等新兴网络形态及应用的飞速发展, 联网设备数量及产生的网络数据流量呈现爆炸式增长, 最先进的核心交换机也已经达到每秒数千 Tb (terabits) 的数据交换带宽, 使得网络流量测量必须利用网络设备中极其受限的高

引用格式: 卜霄菲, 黄河, 孙玉娥, 等. 基于自适应 Sketch 的高速网络流大小测量机制. 中国科学: 信息科学, 2024, 54: 1677–1691, doi: 10.1360/SSI-2023-0294
Bu X F, Huang H, Sun Y E, et al. Adaptive Sketch: accurate flow size measurement in high-speed networks (in Chinese). Sci Sin Inform, 2024, 54: 1677–1691, doi: 10.1360/SSI-2023-0294

© 2024《中国科学》杂志社

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

速处理资源实时抓取并处理隐藏在海量高速网络流中的流量信息, 高效地测量每条流的大小, 是一项极具挑战的工作. 高速网络流实时测量需要在网关或路由器端对高速网络流数据实时处理, 抓取并记录其中的有用信息. 为了匹配网络流速, 网关或路由器必须以极快的速度处理和转发数据包 [1], 只有更新速度极快的网络处理芯片的片上存储资源 (如静态随机存取存储器 SRAM (static random-access memory)) 才能满足数据包的实时存储需求. 然而, 受工艺限制, 此类存储器造价昂贵且规模极度受限, 通常只有 10~30 MB 左右, 且需要与路由、安全、资源调度、质量控制等多个交换设备必须涵盖的功能模块共享, 能够分配用于网络数据实时处理的空间大小往往不超过 1 MB, 难以满足网络流量测量模块对数据实时且持续的超高存储需求 [2, 3]. 概率数据结构 (Sketch) 是一类高效紧凑压缩型数据结构方法的总称, 具有强压缩的特点, 有助于解决当前高速网络实时测量所面临的高速存储资源极度受限的困境 [4, 5]. 因此, 已有大量高速网络流大小测量研究将整个实时数据存储结构放置在片上, 采用最小计数 (Count-Min) [6]、Counter Braids [7] 等计数器级的流共享技术实现流的实时压缩存储, 使不同流共享同一组计数器, 并通过取最小噪声计数器、联立解线性方程等方式尽可能地去除噪声, 以提高估计精度. 例如, HeavyKeeper [8] 放弃了对小流信息的存储, 选择通过指数衰减策略在有限的存储空间下尽可能地保留大流的流标签和统计信息. Pyramid Sketch [9] 提出了一种金字塔型的多层计数器数组架构, 能够在编码更新阶段根据到达流的大小为其提供适应规格的计数器, 该机制通过减少大小流间的噪声干扰显著提高了估计结果的准确性. PR-Sketch [10] 则在流大小查询阶段用片下存储记录的流标签还原各条流在片上计数器数组内的哈希映射关系, 通过求解表示流大小和计数器数值关系的线性方程组实现了一个近似零相对误差的测量机制.

然而, 高速网络流的规模分布极不均衡, 其中绝大多数流为极小流, 例如数据包总数小于 40 的流占比超过 90%. 对于实际的网络测量需求来说, 准确测量极小流 (mouse flow) 规模的实际意义有限, 但却会占用大量稀缺的片上存储空间. 此外, 由于数据包所属流的规模很难在其实时到达时被准确判断, 这使得我们必须预先为每条流开辟足够大的计数空间以满足网络中极少数超大流的测量需求, 大量分配给小流的存储空间导致了严重的资源浪费. 面对该问题, 现有研究大都将不同流的信息混杂存储在一起以提高片上存储的利用率, 但却额外引入了大量难以高效去除的噪声并增加了计算开销, 使得中小流难以达到一个可接受的估计精度. 针对上述问题, 本文研究如何设计一种能够根据流大小自适应调节所占用存储空间的 Sketch 结构, 使其能够高效地滤除小于给定阈值的流, 即滤除对实际应用意义较小的超小流, 并控制超大流过多资源占用, 从而获得低存储成本的高精度流大小测量机制. 所设计的自适应 Sketch 首先利用可逆计数器以极小的空间代价实现对绝大多数极小流的高效滤除, 大幅降低海量噪声小流对网络测量的影响; 然后, 设计了一组采样频率逐层递减的采样计数器, 实现对不同规模流的自适应采样, 从而保证不同规模的流均被高精、低开销地测量, 从而满足不同规模流的差异化测量需求. 通过在真实网络数据集上的测试发现, 基于自适应 Sketch 所提出的流大小估计器在极低的片上存储开销下实现了对大流的精确估计, 特别地, 针对中小规模流的大小估计, 其相对于同类方法展现出了更低的平均相对误差水平. 本文的主要创新点如下.

- 创新性地提出了一种能根据流大小自适应分配存储空间的自适应 Sketch 技术. 该技术利用可逆计数器高效滤除高速网络中的海量噪声小流, 并利用由一组采样频率逐层递减的采样计数器实现对不同规模流的自适应采样, 以尽可能小的存储代价满足不同规模流对估计区间和估计精度的要求.
- 提出了一个高效的高速网络流大小估计器, 利用所设计的自适应 Sketch 实现了实时的噪声小流滤除和自适应流量抓取, 结合概率分析实现了低存储、计算开销的高精度每流估计.
- 利用 CAIDA 2019 提供的真实网络数据对所设计的流大小估计机制进行了实验验证, 实验结果表明, 基于自适应 Sketch 的流大小估计器仅需要极小的片上存储空间就能够实现对不同规模流的

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

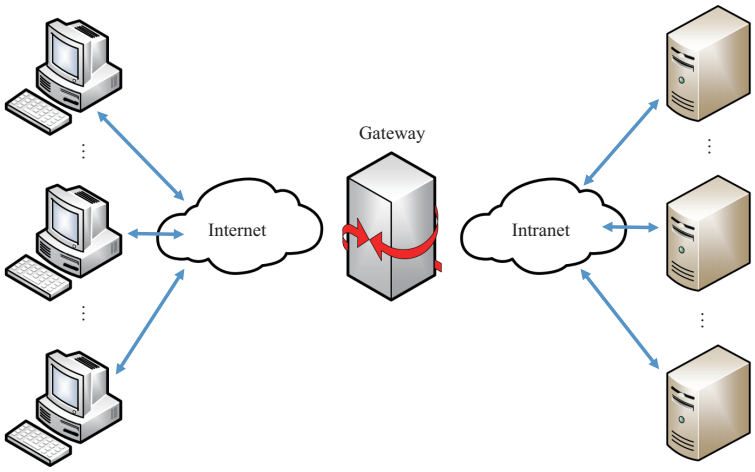


图 1 (网络版彩图) 高速网络流量测量模型

Figure 1 (Color online) Traffic measurement model in high-speed networks

精确估计. 在 22.75 kB 的片上存储空间条件下, 自适应 Sketch 在流大小估计方面的平均相对误差较 Count-Min 方法降低了 76%, 而相对于 virtual HyperLogLog 方法则下降了约 92%.

2 网络测量模型与设计目标

2.1 网络测量模型

本文所研究的高速网络流量测量模型如图 1 所示, 由一个中心网关、内网和外网 3 部分组成, 其中, 网络流量测量估计器作为软件或硬件安装在中心网关上, 用于监控通过该网关的所有网络流. 假设内网由一组服务器组成, 外网由一组客户机组成. 每当有数据包到达时, 中心网关在转发前首先提取数据包中携带的流标签, 并将该数据包记录为所属流开辟的估计空间中, 用于后续的流大小测量. 在这里, 流标签是一个预定义的包头信息组合, 可以根据实际应用需求将关键字、目的地址、源地址、端口号等信息抽象为流标签. 而所抽象出的一条流则是由所有携带相同流标签的所有数据包组成, 流大小测量的目标则是估计每条流所包含的数据包数量.

用 F 表示中心网关所监测到的流集合, 用 N_f 表示流 $f \in F$ 的大小, 用 $\hat{N}_f$ 表示 N_f 的估计值. 所设计的流大小测量模块采用片上和片下相结合的方式. 在每个测量周期, 流量测量模块在网络处理器芯片的片上存储空间开辟一个大小为 m 的物理空间用于快速滤除噪声小流信息、高效记录所监测流的流量信息等. 然后, 在测量周期结束后将得到的流量记录下载到空间不受限的片下空间, 用于后续的分析以估计得到每条流的 $\hat{N}_f$ 值.

2.2 设计目标

本文致力于设计一种更为高效的流大小估计器, 利用极度受限的片上存储资源, 针对流大小分布极不均衡的情况, 提出一种新的 Sketch 结构, 使其能够高效滤除混杂在高速网络流中的海量噪声小流, 并能够根据流的大小自适应为其分配存储空间, 从而解决现有研究工作所存在的局限性, 大幅提升流大小估计精度. 所设计的流大小估计器应满足以下设计目标.

• **片上空间占用小.** 所设计的估计器应在占用尽可能少的片上高速缓存的前提下, 保留尽可能多的有效流量信息. 因此, 所设计的自适应 Sketch 结构应能够为不同规模的流分配合适的存储空间, 以

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

尽可能地提高存储效率.

• **计算复杂度低.** 为了匹配流速, 网络处理器芯片通常仅能支持对每个数据包执行有限的操作, 因此所设计的流大小估计器应具有尽可能少的片上计算复杂度, 以保证所有数据包能够被实时处理.

• **估计精度高.** 所设计的自适应 Sketch 需要能够满足不同规模流的估计区间需求, 并尽可能地降低由存储共享所带来的噪声影响, 通过严谨的概率分析, 实现不同规模流的高精度流大小估计.

3 自适应 Sketch 设计

3.1 设计思想

高速网络流量分布是典型的长尾分布, 绝大多数网络流是规模较小的网络流, 仅有少部分大流占据了较多的网络资源. 例如在 CAIDA 数据集的一分钟网络数据中, 超过 90% 的网络流是规模小于 40 的超小流, 但该数据集中规模最大的前 0.35% 的网络流产生了超过 50% 的网络流量, 而最大流的规模则达到了约 570 K. 由此产生的问题和困难是: 一方面, 若因难以预知的流大小而事先为每条流分配相同的存储空间来记录流量数据, 就会造成极度紧缺的片上高速缓存资源的浪费, 降低测量系统的整体性能; 另一方面, 网络流量分布的极不均衡, 大量的超小流会引入测量噪声进而导致大流测量误差, 进一步加剧了网络测量的难度. 自适应 Sketch 就是为解决上述问题而设计的, 期望利用极其受限的片上存储资源和少量的片上操作, 高效地滤除混杂在海量网络流中的超小噪声流信息, 并为不同规模的流分配合适的存储空间, 以减少片上存储占用并提高测量精度.

3.2 自适应 Sketch 结构

图 2 展示了自适应 Sketch 的结构示例, 它包括噪声小流过滤和自适应采样计数两个模块, 用于对网络流数据进行实时筛选和记录. 噪声小流过滤模块利用 l_1 层的可逆计数器实现对网络中绝大多数超小流的有效过滤, 降低进入后续采样计数模块的流数量. 随后, 自适应采样计数模块设置 l_2 的采样计数器, 通过逐层降低采样频率的方式, 实现基于流规模的自适应采样, 从而为不同规模的流分配合适的存储空间. 接下来, 我们给出这两种计数器的具体定义及其相应的功能.

定义1 (**可逆计数器**) 可逆计数器是一种带有标志位的计数器, 其中最高位充当标志位. 标志位为 1 时, 表示计数器的值为正数; 反之, 标志位为 0 时, 计数器的值为负数. 可逆计数器在计数时, 可以根据预定规则选择对计数器进行 “+1” 或 “−1” 操作.

自适应 Sketch 结构的噪声小流过滤模块由 l_1 层的可逆计数器组成, 其中, 第 $i \in [1, l_1]$ 层包含 m_i 个大小为 d_i 比特的可逆计数器. 在进行流量测量时, 由于海量网络流的分布极不均匀, 通常需要多条流共享同一块物理内存进行测量. 通过让共享同一可逆计数器的流以一定概率选择 “+1” 或 “−1” 操作, 从而使大量噪声小流的流量因相反操作而相互抵消. 最终, 可逆计数器的值趋近于主导流 (即共享该可逆计数器的最大流) 的值. 为防止多个大流共享同一计数器而导致相互抵消, 每个流的流量将均匀映射到 k 个计数器上, 并在每个计数器上以概率独立选择操作. 以图 2 所示为例, 对于流 f 到达的每个数据包, 依据流标签映射到 3 ($k = 3$) 个计数器, 然后均匀随机选择其中一个计数器, 并在该计数器上独立选择 “+1” 或 “−1” 操作.

定义2 (**采样计数器**) 采样计数器是一个预设采样频率与一个计数器的组合. 采样计数器在计数时会按照该计数器预设的采样概率进行加一操作, 只记录被采样元素的个数.

自适应采样计数模块由 l_2 层的采样计数器组成. 对于第 $j \in [1, l_2]$ 层, 包含 m_j 个不带标志位、大

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

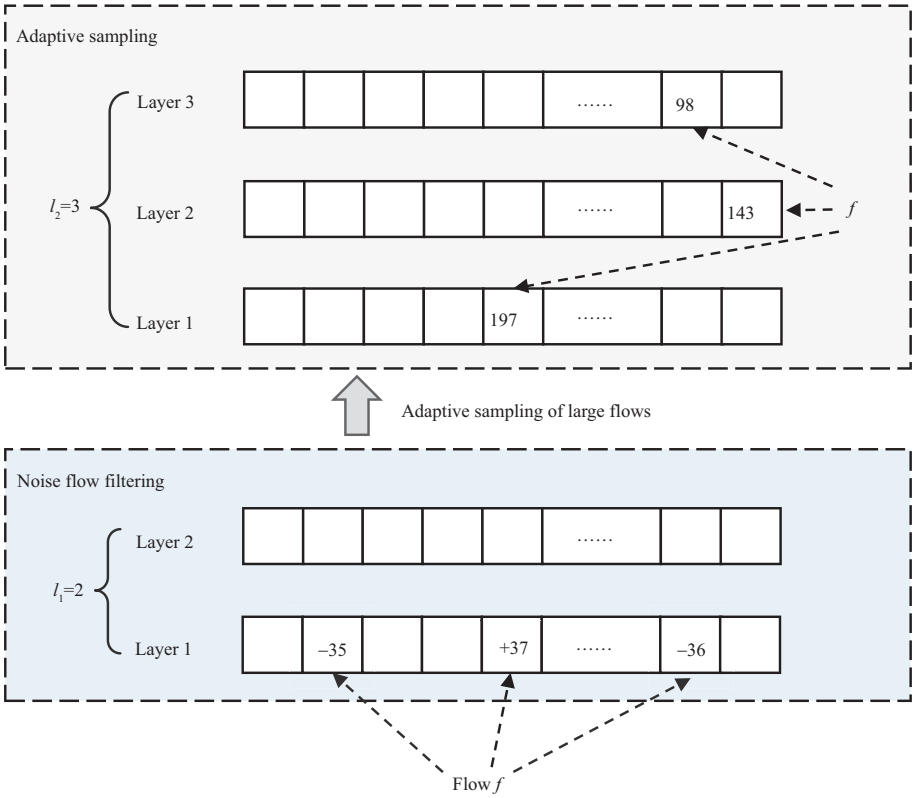


图 2 (网络版彩图) 自适应 Sketch 的结构与示例

Figure 2 (Color online) Data structure and example of Adaptive Sketch

小为 d_j 比特的采样计数器, 以及预设的采样概率 p_j. 在流大小测量中, 不同规模的网络流所需的计数器大小不同, 所能容忍的数据丢失率, 即能接受的最小采样频率也不同. 一般来说, 规模越大的网络流所需开辟的计数器位数越大, 所能容忍的采样频率越小. 通过设置逐层递减的采样频率, 仅将采样后的流量记录在计数器中, 有助于在保证不同规模流的测量精度的同时降低存储资源的占用. 以图 2 所示为例, 流 f 的数据包进入采样计数模块后, 在每一层, 都以该层预设的采样概率进行采样. 若数据包在某一层被成功采样, 则利用哈希函数映射到该层的计数器上, 并对其计数值 +1. 对于大规模的流, 低层的采样计数器可能无法保证满足其计数区间需求, 较高的采样频率可能会导致计数器溢出, 从而使得基于该计数器的估计值出现较大误差. 为了解决这一问题, 在每一层上, 每条流的流量都会被均匀分散到 k 个采样计数器上, 避免流规模超出计数范围, 以及部分流因与大流哈希碰撞共享同一计数器而导致估计误差较大问题. 通过这一设计, 规模相对较小的流能够通过低层具有更高采样频率的计数器估计流大小, 而超大规模的流因为可以容忍更小的采样频率, 从而在高层的采样计数器中进行精确记录. 最终, 实现基于流规模自适应采样的效果, 在保证估计精度的同时尽可能降低存储开销.

4 基于自适应 Sketch 的流大小测量

基于自适应 Sketch 结构, 我们提出了一个流大小测量方案, 涵盖了片上实时噪声小流过滤、自适应采样的实时流量抓取以及片下离线流大小估计 3 个部分. 在测量周期的初始阶段, 我们在片上内存空间中部署一个自适应 Sketch 结构, 用于记录每个测量周期内的网络流量. 具体而言, 噪声小流过滤

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

模块在此阶段负责实时过滤网络流中的大量噪声小流, 而自适应采样模块则实时抓取每条流的流量信息, 实现基于流大小的自适应采样记录. 在单个测量周期结束后, 我们将记录流量信息的自适应 Sketch 结构下载到片下, 用于进行流大小的离线估计. 同时, 为确保下一周期的准确性, 我们将片上的自适应 Sketch 恢复为初始状态. 接下来, 详细介绍所提出的流大小测量方案的各个部分.

4.1 片上噪声小流过滤机制

在每个测量周期的初始阶段, 所有计数器的值均初始化为 0. 对于每一个实时到达的数据包, 首先计算下面的哈希输出:

$$h_b = H(f \oplus C[H(b) \bmod k]), \tag{1}$$

其中, H 表示一个哈希函数, b 是该数据包的 ID, f 是数据包 b 所属的流标签, C 是一个包含 k 个随机种子的数组. 设 n_i 表示第 $i \in [1, l_1]$ 层可逆计数器的数量, 通过进一步计算 $h_b \bmod n_i$ 可以确定数据包 b 散列到第 i 层可逆计数器的索引. 此外, 依据式 (1) 有, h_b 的取值范围为 $\{H(f \oplus C[0]), H(f \oplus C[1]), \ldots, H(f \oplus C[k-1])\}$, 换言之, 同属于流 f 的所有数据包将被均匀随机地映射到该流在第 i 层的 k 个可逆计数器中. 以图 2 中 $k = 3$ 的情况为例, 在到达第 1 层时, 流 f 的所有数据包被均匀地散列到该层的 3 个可逆计数器上.

可逆计数器在计数时根据预定规则对计数器进行 "+1" 或 "−1" 操作. 为了满足单一流的计数需求, 必须确保在选择操作时, 同一条流的数据包在同一计数器上具有相同的操作选择. 为此, 我们设置每条流以概率 P' 独立选择在可逆计数器上的 "+1" 操作, 以 $1 - P'$ 的概率选择 "−1" 操作以记录自身的流量. 具体来说, 针对流 f 的任意数据包 b, 计算随机值 R_f:

$$R_f = \frac{H(f \oplus (h_b \bmod n_i)) \bmod M}{M}, \tag{2}$$

其中, $(h_b \bmod n_i)$ 表示数据包 b 在第 i 层上的计数器索引. M 为不超过该层计数器数量的最大素数, 即 $M \leqslant n_i$. 若 $R_f < P'$, 流 f 的数据包 b 在计数器 $(h_b \bmod n_i)$ 上进行 "+1" 操作; 否则, 执行 "−1" 操作. 例如, 在图 2 中, 流 f 的数据包分别在 2 个计数器上执行 "−1" 操作 (对应值为 $-35, -36$ 的两个计数器) 和 1 个 "+1" 操作 (对应值为 $+37$ 的计数器).

数据包 b 在映射到第 i 层可逆计数器 $(h_b \bmod n_i)$ 后, 首先判断该可逆计数器是否已满. 若计数器未满, 则根据式 (2) 对该计数器执行相应操作; 若计数器已满, 则判断计数器的当前取值与操作选择是否相反, 若相反, 则直接丢弃数据包 b; 否则, 溢出至下一层可逆计数器. 在这里, 计数器的标志位为 1, 表示该计数器为正值, "−1" 操作被定义为与计数器的正值相反; 相应地, "+1" 操作被定义为与计数器的负值相反. 对于流 f 从第 i 层溢出的数据包, 将进入第 $i+1$ 层可逆计数器, 通过计算 $(h_b \bmod n_{i+1})$ 将数据包 b 映射到指定的计数器上, 并进行与第 i 层相似的操作流程. 依次类推, 直到流 f 在所有 l_1 层的可逆计数器均满为止, 此时流 f 仍溢出的数据包将进入自适应采样计数模块.

4.2 自适应采样流量实时抓取机制

采样计数模块的主要设计目标是利用采样计数器实现不同规模流的自适应采样计数, 以降低存储开销. 在该模块, 每层部署一组不带标识位的计数器, 并设置一个相应的采样概率. 设第 $j \in [1, l_2]$ 层的采样概率为 p_j. 对于每个进入采样计数模块的数据包, 从第 1 层到第 l_2 层逐层采样并进行后续的处理. 设流 f 的数据包 b 在第 j 层上被成功采样, 通过计算式 (3) 确定数据包 b 所映射到的采样计数器索引 h.

$$h = H(f \oplus C[H(b) \bmod k]) \bmod n_j, \tag{3}$$

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

其中 H 表示哈希函数, f 为数据包 b 所属的流标签, C 是一个包含 k 个随机种子的数组, n_j 为第 j 层采样计数器的数量. 若索引为 h 的采样计数器未达到其最大计数值, 则计数器加一以记录流量信息; 否则, 数据包不做处理. 通过式 (3) 可以发现, 流 f 所有进入采样计数模块的数据包也将被均匀随机地分散到第 j 层的 k 个采样计数器中, 这可以降低流之间的计数器共享, 从而提供更准确的流估计结果. 此外, 采样概率逐层递减的设置也使得自适应 Sketch 在无需预测流量规模的基础上, 即可实现基于流规模的自适应采样.

4.3 片下离线流大小估计

在每个测量周期结束后, 中心网关会将自适应 Sketch 中记录的流量信息下载到片下, 用于后续的离线估计. 设 N_f 为流 f 的流量大小, $\hat{N}_f$ 为 N_f 的估计值. 对于每条流 f, 其流量在经过噪声小流过滤模块后进入采样计数模块, 因此, 针对流 f 大小的估计可以分为两步: (1) 估计流 f 进入采样计数模块的流量大小, 记为 $\hat{N}_f^s$; (2) 估计流 f 的流量大小, 即 $\hat{N}_f$.

(1) 估计流 f 进入采样计数模块的流量大小 $\hat{N}_f^s$. 采样计数模块中, 流 f 的每个数据包分别在各层依据指定的采样概率进行采样, 采样成功后将被哈希到指定的计数器记录. 此外, 在每一层上, 流 f 所有成功采样的数据包将被均匀分散到该层 k 个计数器中. 基于此, 我们可以估计流 f 进入采样计数模块的流量大小, 即 $\hat{N}_f^s$ 的值. 首先, 找到流 f 在所有采样计数器中的最小值, 因为最小的计数器意味着最少的噪声干扰. 假设流 f 最小计数器的值为 min, 且所在的层数为 $j \in [1, l_2]$, 则流 f 在第 j 层上记录的流量约为 $\min \times k$. 而在第 j 层上, 流 f 以 p_j 的概率进行采样, 因此, 流 f 进入采样计数模块的流量大小约为

$$\hat{N}_f^s = \frac{\min \times k}{p_j}. \tag{4}$$

(2) 估计流 f 的流量大小 $\hat{N}_f$. 在自适应 Sketch 的设计中, 流 f 的数据包在经过 l_1 层的可逆计数器后进入采样计数模块. 通过建立 $\hat{N}_f$ 与 $\hat{N}_f^s$ 之间的联系, 可以求解出流 f 的估计大小 $\hat{N}_f$. 接下来, 我们以本文实验所设置的 $l_1 = 2$ 为例求解 $\hat{N}_f$ 的值, 对于 $l_1 > 2$ 的情况类比可得. 设 $\hat{N}_f'$ 为流 f 经过第 1 层可逆计数器后的流量大小, 则 $\hat{N}_f'$ 的值表示为

$$\hat{N}_f' = \hat{N}_f \cdot p \cdot \frac{k_1}{k} - \alpha \cdot k_1 \cdot \mathrm{Th}_1, \tag{5}$$

其中, p 表示自适应 Sketch 在噪声小流过滤模块之前的采样概率, 通过调整 p 的值, 自适应 Sketch 可以适配不同规模的网络流量. 在每一层可逆计数器上, 流 f 的流量会均匀的散列在 k 个计数器中, 且以概率 P' 独立选择在计数器上的 "+1" 操作, 以概率 $1 - P'$ 选择 "−1" 操作. $k_1 \in [0, k]$ 则表示流 f 在第 1 层的 k 个可逆计数器中, 取值与流 f 的操作选择一致的计数器数量. Th_1 表示第 1 层可逆计数器的计数区间, 例如, 对于 8 比特带标志位计数器的计数区间为 $2^7 - 1$. $\alpha \in [0, 1]$ 为系数.

流 f 在第 1 层溢出的流量将会经过第 2 层可逆计数器, 并采用相同的方式记录流量, 区别在于, 在第 2 层前不再以概率 p 进行采样. 设 $\hat{N}_f''$ 为流 f 经过第 2 层可逆计数器后的流量大小, 通过替换式 (5) 中的变量, 可以得到 $\hat{N}_f'$ 与 $\hat{N}_f''$ 之间的等式关系, 如下所示:

$$\hat{N}_f'' = \hat{N}_f' \cdot \frac{k_2}{k} - \alpha \cdot k_2 \cdot \mathrm{Th}_2, \tag{6}$$

其中, $k_2 \in [0, k]$ 表示流 f 在第 2 层的 k 个可逆计数器中, 取值与流 f 的操作选择一致的计数器数量. Th_2 表示第 2 层可逆计数器的计数区间.

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

表 1　自适应 Sketch 的参数设置

Table 1　Parameter settings for Adaptive Sketch

Layer	Type	Size (bits)	k	Sampling probability	P'
1	Reversible counter	8	3	0.5	0.5
2	Reversible counter	8	3	–	0.5
3	Sample counter	13	3	0.5	–
4	Sample counter	13	3	0.25	–
5	Sample counter	13	3	0.1	–

在噪声小流过滤模块只有两层可逆计数器 ($l_1 = 2$) 的情况下, $\hat{N}_f''$ 的值就等于 $\hat{N}_f^s$ 的值. 因此, 通过联立式 (4)∼(6), 可以得到流 f 的流量大小 $\hat{N}_f$ 的估计公式, 即

$$\hat{N}_f = \frac{k}{p \cdot k_1} \cdot \left(\frac{\min \cdot k^2}{k_2 \cdot p_i} + \alpha k_1 \mathrm{Th}_1 + \alpha k \mathrm{Th}_2 \right). \tag{7}$$

5　实验

本文实验实现在一个具有 12 核心的服务器 (Intel(R) Xeon(R) CPU E5-2643 v4 3.40 GHz) 上, 该服务器具有 20 MB 缓存、256 G RAM. 为测试自适应 Sketch 在真实流量上表现, 我们使用 2019 年 CAIDA 监听的匿名流量数据作为流大小估计的数据集, 并与经典算法 Count-Min 和 virtual HyperLogLog (vHLL) 进行对比. 实验结果表明自适应 Sketch 方法不仅在大流估计上具有最优的估计精度, 而且针对中小规模流的估计, 其相对于同类方法也具有更低的平均相对误差水平. 本节将首先介绍相关的实验参数设置, 然后对实验结果进行详细的分析.

5.1　参数设置

自适应 Sketch 定期将片上记录的流量数据传输到片下, 以进行流量大小的估计. 在不失一般性的情况下, 我们选取一分钟作为测量周期的基准. 为了排除数据集中潜在的特殊性, 我们将测试范围扩展至 5 个周期的流量数据, 并对结果进行平均处理, 以消除可能由于特殊情况引起的测试误差. 我们以目标地址作为流标签, 该测试数据共包括 147 M 个数据包, 486 k 条流, 平均每分钟 30 M 个数据包, 97 k 条流. 在这个基础上, 对自适应 Sketch 进行了一系列实验, 测试其在不同参数组合下的流量大小估计效果.

本实验所使用的自适应 Sketch 结构与图 2 中展示的相同. 其中, 第 1 和 2 层构成噪声小流过滤模块, 用于过滤噪声小流; 第 3∼5 层组成采样计数模块, 用于记录大流的信息, 即 $l_1 = 2$, $l_2 = 3$. 其他自适应 Sketch 的参数, 包括计数器大小、采样概率等, 详见表 1. 同时, 表 2 提供了自适应 Sketch 各层计数器的数量以及对应的片上内存占用. 这样设置旨在测试各模块计数器数量变化对自适应 Sketch 性能的影响.

5.2　实验结果

图 3∼5 展示了在不同片上内存空间下自适应 Sketch 算法的流量大小估计结果, 同时与 virtual HyperLogLog 和 Count-Min 算法进行了比较. 在这些图中, 第 1 层和第 2 层可逆计数器的数量分别为 2000 和 1800, 采样计数器数量与片上内存大小的对应关系见表 2. 图中的每个点代表一条网络流, 横坐标表示流的真实大小, 纵坐标表示流的估计大小, 图中的点越靠近 $y = x$ 这条线, 表示算法的估计

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

表 2 各层计数器数量及片上内存占用

Table 2 Number of counters and on-chip memory usage

Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Memory (kB)
2000	1800	3000	2000	1000	13.23
2000	1800	4000	3000	2000	17.99
2000	1800	5000	4000	3000	22.75
2500	2000	3000	2000	1000	13.92
2500	2000	4000	3000	2000	18.68
2500	2000	5000	4000	3000	23.44
3000	2000	3000	2000	1000	14.40
3000	2000	4000	3000	2000	19.17
3000	2000	5000	4000	3000	23.92

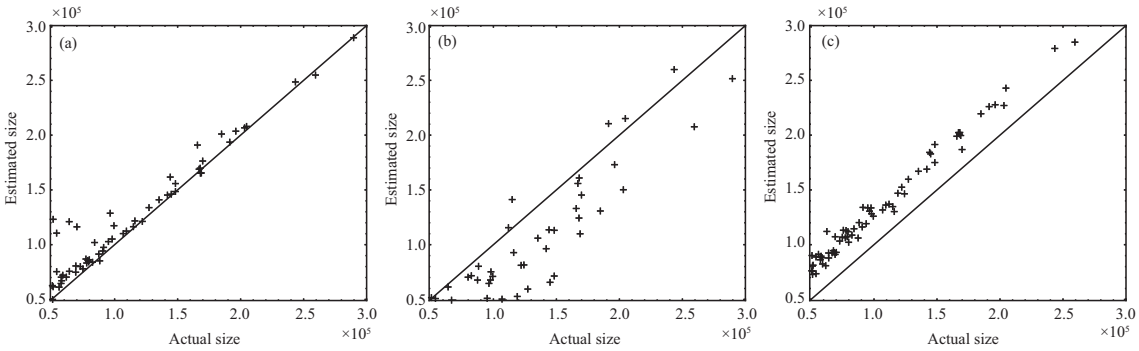


图 3 13.23 kB 片上内存空间下各算法的流大小估计

Figure 3 Estimation of flow sizes for various algorithms with 13.23 kB on-chip memory. (a) Adaptive Sketch; (b) virtual HyperLogLog; (c) Count-Min

越准确, 因为 $y = x$ 代表了估计值等于真实值的理想情况. 在相同的片上内存空间下, 自适应 Sketch 算法相对于其他两种算法展现出更为出色的估计效果. 以图 5 为例, 自适应 Sketch 对流大小的估计结果与真实值非常接近. 即使对于不同规模的流, 自适应 Sketch 也能够提供准确的估计结果. 相比之下, virtual HyperLogLog 算法在流大小估计方面表现不尽如人意. 其估计结果分布在 $y = x$ 线的两侧, 表明该方法在不同流大小的估计上存在较大的不一致性, 而在内存空间较小的情况下, 这种估计误差将更加显著. 相对而言, Count-Min 算法的性能要优于 virtual HyperLogLog 算法, 并且随着内存大小的增加, Count-Min 算法的估计结果也有明显的改善. 然而, 该算法仍然存在一个显著问题, 即高估流量大小. 这是由于大量的噪声小流对计数器的共享所引起的, 导致了估计结果的偏差.

通过比较图 3(a), 4(a) 和 5(a), 我们可以观察到, 随着片上内存空间的增加, 自适应 Sketch 算法在流大小估计方面的准确性逐渐提升. 图 3(a) 展示了在片上内存空间为 13.23 kB 的情况下自适应 Sketch 的估计结果. 在这种情况下, 许多中等规模的流的估计结果高于其真实值, 甚至有个别流的估计出现了严重的偏差. 然而, 当内存增加到 17.99 kB (图 4(a)) 时, 大多数流的估计结果贴近其真实大小, 只有一些中小规模的流存在较大误差. 当内存空间进一步增加到 22.75 kB (图 5(a)) 时, 中等规模及以上的流的估计大小都近似于其真实大小. 更值得鼓舞的是, 中小规模流的估计结果也更加准确, 表现出随着片上内存增加而改善的趋势. 这说明, 在基于可逆计数器进行噪声小流过滤的前提下, 更大的

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

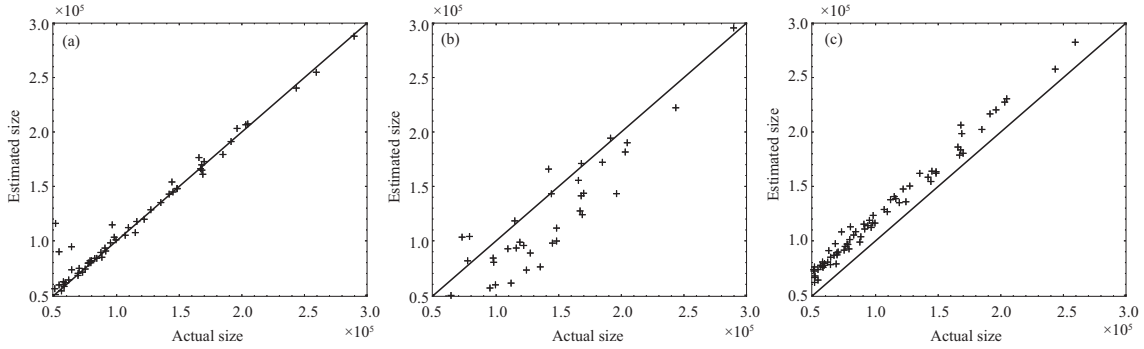


图 4　17.99 kB 片上内存空间下各算法的流大小估计

Figure 4　Estimation of flow sizes for various algorithms with 17.99 kB on-chip memory. (a) Adaptive Sketch; (b) virtual HyperLogLog; (c) Count-Min

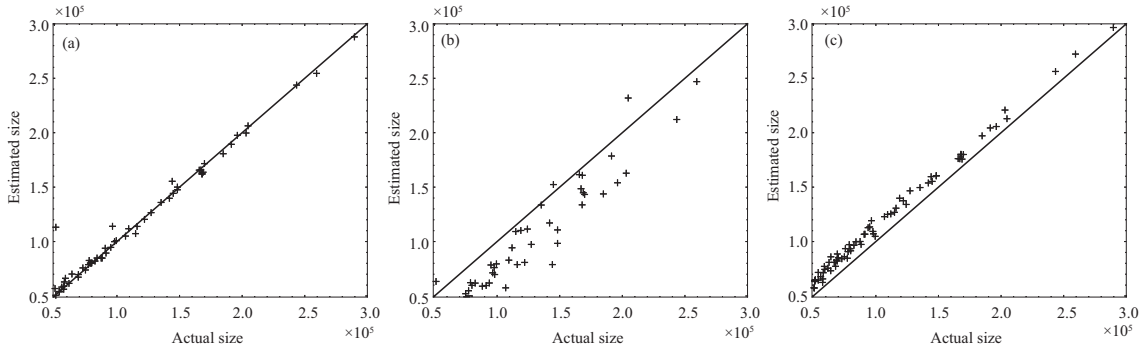


图 5　22.75 kB 片上内存空间下各算法的流大小估计

Figure 5　Estimation of flow sizes for various algorithms with 22.75 kB on-chip memory. (a) Adaptive Sketch; (b) virtual HyperLogLog; (c) Count-Min

片上内存空间使得算法能够配置更多的采样计数器, 进一步减少了流之间计数器的共享, 从而提高了估计的精确性. 这些发现强调了采样计数器数量对自适应 Sketch 算法性能的重要影响.

为了量化自适应 Sketch 算法在流大小估计方面的综合表现, 我们进一步计算该方法的平均相对误差. 平均相对误差 (average relative error, ARE) 是一个用于衡量估计结果与真实值之间差异的统计指标, 在流量测量的上下文中, 可以定义为流大小的估计值与流真实大小之间的相对差异, 它通过以下公式计算:

$$\mathrm{ARE} = \frac{1}{n}\sum_{f=1}^{n}\left|\frac{N_f - \hat{N}_f}{N_f}\right|, \tag{8}$$

其中, N_f 表示流 f 的真实大小, $\hat{N}_f$ 表示流 f 的估计大小, n 表示一个测量周期内流的总数. ARE 的数值越小, 表示流大小估计方法的估计结果与流真实大小的吻合度越高.

针对表 2 中各层计数器的数量设置, 我们分别对每个计数器数量下的自适应 Sketch 计算了 ARE, 并将结果展示在图 6 中. 具体来说, 在图 6(a) 中, 我们研究了第 1 和 2 层可逆计数器分别设置为 2000 和 1800 时, 不同采样计数器数量对 ARE 的影响. 图 6(b) 则考察了第 1 和 2 层可逆计数器分别设定为 2500 和 2000 时的 ARE 表现, 图 6(c) 对应第 1 和 2 层可逆计数器分别设定为 3000 和 2000 时的 ARE 情况.

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

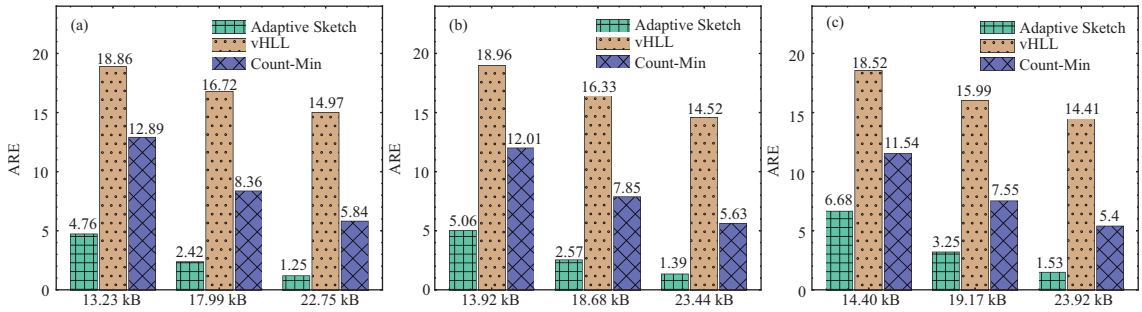


图 6 (网络版彩图) 自适应 Sketch 在不同计数器数量下的 ARE

Figure 6 (Color online) ARE of Adaptive Sketch under different number of counters. (a) Layer 1 = 2000, Layer 2 = 1800; (b) Layer 1 = 2500, Layer 2 = 2000; (c) Layer 1 = 3000, Layer 2 = 2000

首先, 我们分析增加采样计数器对自适应 Sketch 算法 ARE 的影响. 以图 6(c) 为例, 当可逆计数器数量确定为 3000 和 2000, 采样计数器的数量从 3000, 2000, 1000 (占用片上内存 14.40 kB) 增加到 5000, 4000, 3000 (占用片上内存 23.92 kB) 时, 自适应 Sketch 算法的 ARE 从 6.68 下降至 1.53, 下降幅度达到了 77%. 这种降低的原因在于增加采样计数器的数量减少了采样计数模块中流之间的计数器共享, 由于减少了由于哈希碰撞所引入的噪声, 进一步提高了流大小估计的准确性, 强调了采样计数器数量增加对提升估计精度的重要性.

然而, 与采样计数器不同, 固定采样计数器并增加可逆计数器的数量并不一定确保自适应 Sketch 方法的 ARE 降低. 例如, 当采样计数器的数量被固定为 3000, 2000, 1000 时, 将可逆计数器的数量从 2000 和 1800 (图 6(a) 中 13.23 kB) 增加到 3000 和 2000 (图 6(c) 中 14.40 kB), 自适应 Sketch 算法的 ARE 从 4.76 增加到了 6.68. 这一现象的解释在于仅增加可逆计数器的数量会降低小流过滤模块中流之间的计数器共享情况. 结果, 一部分小流无法通过共享机制相互抵消被拦截在可逆计数器中, 而会进入采样计数器, 从而增加了对中等规模及以上流估计的干扰. 这种情况强调了在调整算法参数时需要综合考虑不同模块之间的相互影响, 以实现最佳性能.

通过计算自适应 Sketch 算法在不同参数设置下的 ARE, 我们量化了该方法在网络流量大小估计方面的综合表现. 为了更深入地评估自适应 Sketch 算法对不同规模流的估计效果, 我们计算了该方法在不同流大小范围内的 ARE, 并将其与 virtual HyperLogLog (vHLL) 和 Count-Min 方法的结果进行了比较. 实验结果展示在表 3~5 中, 分别对应了片上内存空间为 13.23, 17.99 和 22.75 kB 的情况.

通过实验结果, 可以观察到随着片上内存空间的增加, 3 种方法在各个流大小范围内的 ARE 都呈现出明显的下降趋势. 然而, 与 virtual HyperLogLog 和 Count-Min 算法相比, 自适应 Sketch 方法在不同规模的流大小估计方面表现出了最佳性能. 具体而言, 针对流大小在 10 k 以上的流, 3 种估计算法都展现出了较低的误差水平, 但相较而言, 自适应 Sketch 一直保持着最低的平均相对误差, 且随着片上内存空间的增加, 其平均相对误差逐渐减小.

对于流大小在 10 k 以下的情况, 自适应 Sketch 的 ARE 远小于其他两种估计算法. 例如, 针对大小在 6 k~8 k 之间的流, 在 13.23 kB 的片上内存空间下 (表 3), 自适应 Sketch 算法的 ARE 为 1.944, Count-Min 算法的 ARE 约为其 2 倍, virtual HyperLogLog 则大约是其 3 倍. 特别是对于大小在 1 k~2 k 范围内的流, 自适应 Sketch 只需要极少的片上存储空间, 即可将 ARE 控制在个位数水平. 如表 3 所示, 自适应 Sketch 的 ARE 仅为 8.686, 约为 Count-Min 算法的 1/3, virtual HyperLogLog 的 1/4. 当片上存储空间增大至 22.75 kB (表 5) 时, 自适应 Sketch 算法的 ARE 下降至 2.228, 下降幅度

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

表 3 13.23 kB 片上内存下各算法针对不同规模流估计的 ARE 比较

Table 3 Comparison of ARE for different flow size estimations under 13.23 kB on-chip memory

Algorithm	1 k∼2 k	2 k∼4 k	4 k∼6 k	6 k∼8 k	8 k∼10 k	> 10 k
Adaptive Sketch	8.686	4.527	2.496	1.944	1.438	1.010
vHLL	32.576	15.885	9.287	6.403	4.766	2.191
Count-Min	22.201	10.929	6.304	4.279	3.425	1.621

表 4 17.99 kB 片上内存下各算法针对不同规模流估计的 ARE 比较

Table 4 Comparison of ARE for different flow size estimations under 17.99 kB on-chip memory

Algorithm	1 k∼2 k	2 k∼4 k	4 k∼6 k	6 k∼8 k	8 k∼10 k	> 10 k
Adaptive Sketch	4.517	2.129	1.201	1.042	0.798	0.589
vHLL	29.090	13.903	7.968	5.577	4.289	1.920
Count-Min	14.426	7.066	4.040	2.820	2.194	1.033

表 5 22.75 kB 片上内存下各算法针对不同规模流估计的 ARE 比较

Table 5 Comparison of ARE for different flow size estimations under 22.75 kB on-chip memory

Algorithm	1 k∼2 k	2 k∼4 k	4 k∼6 k	6 k∼8 k	8 k∼10 k	> 10 k
Adaptive Sketch	2.228	1.004	0.672	0.685	0.626	0.487
vHLL	25.911	12.666	7.197	4.951	3.810	1.701
Count-Min	10.069	4.946	2.846	1.953	1.498	0.718

为 74%. 同等情况下, Count-Min 的 ARE 下降至 10.069, 下降幅度约为 55%, 而 vHLL 则展现出了较差的性能, 其 ARE 为 25.911, 下降幅度仅约为 20%. 这说明, 自适应 Sketch 算法不仅对大流具有优秀的估计精度, 也对中小规模的流具有良好的估计效果. Count-Min 算法通过增加存储空间, 也能实现对中小规模流的准确估计. 然而, virtual HyperLogLog 算法则仅适合对大流进行估计. 这些实验结果突显了自适应 Sketch 方法在流大小测量方面的准确性和适应性, 尤其是在有限的片上内存空间下, 其相对于同类方法具有更低的平均相对误差水平, 同时所能估计的流大小范围也更加广泛.

在网络流量测量中, 噪声小流对测量结果产生严重负面影响, 因此对小流的高效过滤对于提高流量测量的准确性至关重要. 为了评估自适应 Sketch 在噪声小流过滤方面的性能, 我们进行了对 1 k 以下噪声小流的过滤效果的测试, 并将结果展示在图 7 中. 本实验分别测试了共 5 min 的数据集上的过滤效果, 平均每分钟有约 149038 个 1 k 以下的小流, 自适应 Sketch 在每分钟的过滤中平均成功过滤了约 114022 个小流. 在多个测量周期内, 自适应 Sketch 成功过滤了平均约 76.5% 的 1 k 以下噪声小流. 这种过滤效果不仅能够保留主要流量, 还能够高效过滤掉大量的噪声小流, 从而显著提升后续流量测量任务的准确性. 值得注意的是, 该方法在不同测量周期内都对低阈值小流保持相对稳定的过滤效果, 为实际应用中提供了稳定性和可靠性的保证. 这一特性使得自适应 Sketch 成为网络流量分析中噪声小流过滤的可靠选择.

6 国内外研究现状

流的大小测量, 即测量流中数据包的数量, 可以通过为每条流分配一个计数器来实现. 然而, 高速网络中流的数量过大导致为所有流单独分配计数器是不现实的, 现有机制大都需要多条流共享同一个

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

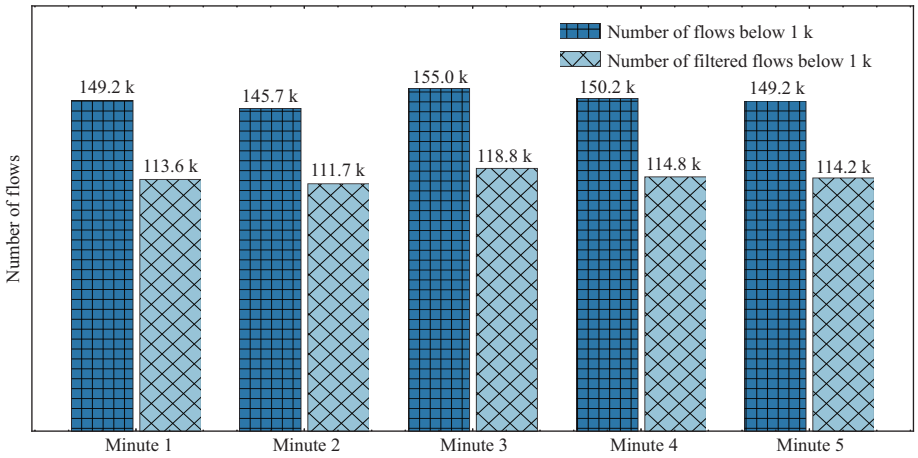


图 7 (网络版彩图) 自适应 Sketch 对于 1 k 以下小流的过滤效果

Figure 7 (Color online) Filtering effect of the Adaptive Sketch on flows below 1 k

计数器. 典型的紧凑型数据结构, 如 Count-Min [6], Counter Tree [11] 和 virtual HyperLogLog [12], 采用流共享技术通过维护一个记录单元 (计数器/寄存器/位) 池, 使不同流共享同一组记录单元, 实现流大小的实时压缩存储. 尽管流共享技术有效提高了记录单元的使用效率, 但这种方式会使得每条流的估计结果存在大量难以去除的噪声. 为了进一步提高估计的准确度, 相关工作, 如 Counter Braids [7], PR-Sketch [10], 为数据结构内的每个计数器建立其所记录的每条流的大小与最终计数结果之间的线性方程, 通过求解所建立的线性方程估计得到每条流的大小. 但此类工作存在计算复杂度过大的问题, 且在共享较多时相关机制很难给出一个收敛的估计结果. 因此, 更大一部分的相关工作把研究重心放在了如何更加合理地使用有限的片上存储资源这一问题上. 考虑到实际的网络流量表现出难以预知以及流大小分布不均的特性, 使得现有紧凑数据结构对大象流的信息存储存在片上资源紧缺, 而对于老鼠流则表现出片上资源浪费的问题. 其中的部分研究放弃了对小流的信息存储, 选择在有限的存储空间下尽可能地保留大流的流标签和统计信息. 例如, Space Saving [13] 算法通过维护一个特定大小的 Stream-Summary 结构, 实时地剔除尾部的小流. HeavyKeeper [8] 通过指数衰减策略, 根据流当前的频数大小赋予其不同的衰减概率 (以指数函数的形式, 越大的流衰减概率越低), 在有限的存储空间下尽可能地保留大流的流标签和统计信息. Zhao 等 [14] 认为现有的紧凑数据结构对于不同规模计数器的配置方案不够灵活, 提出了可以根据当前的流分布动态调整计数单元大小的 DHS 算法. WavingSketch [15] 则为不同的流级测量目标提供了一种通用、无偏的估计方法. 而为了降低采样方案的片上内存占用与通信开销, Du 等 [16] 提出了一个自适应采样框架. 该框架根据流的大小进行自适应概率采样, 通过利用哈希函数, 将真实元素压缩为虚拟元素, 并将通过非重复采样的虚拟元素保存到片下. 为了更有效地平衡大流和小流的压缩效果, 他们引入了两种不同的压缩方式, 分别是线性压缩和对数压缩, 在提高估计精度的同时, 显著降低采样方案的内存和通信负担, 从而提高整体性能. 另一部分工作则尝试通过大小流的差异化存储实现更加合理的存储空间利用, 准确记录网络流量中所有流的统计信息. 例如, ICE Buckets [17] 根据被哈希映射到桶内的流的大小动态地决定当前桶内最优的概率计数方式, 通过桶间计数精度存在差异的测量机制使流的最终估计误差仅与同样被映射到同一个桶的流存在关联. Pyramid Sketch [9] 提出了一种金字塔型的多层计数器数组架构, 在编码更新阶段根据到达流的大小为其提供适应规格的计数器, 通过减少大小流间的噪声干扰显著提高了估计结果的准确性. Stingy Sketch [18] 则通过 BC 树将计数空间分割成树型结构的计数节点, 并结合基于流水线调度的预访存技

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

术在保证估计准确性的同时减少内存访问开销. Cold Filter [19] 和 LogLog Filter [20] 则通过设计一个过滤结构实现数据流中的大小流分离, 并将大流对应的数据包传至后续的数据流测量模块, 实现数据流中大小流的分离测量.

7 总结与展望

本文提出了一种能够根据每条流的大小自适应调整存储策略的 Sketch 技术, 用于解决高速网络测量所面临的高速缓存极度稀缺的挑战, 从而达到提高每流大小测量精度的目的. 基于所提出的自适应 Sketch 技术所设计的每流大小估计器通过极低存储代价高效滤除绝大多数噪声小流, 并通过逐层递减的采样计数器设计, 实现基于流量的自适应采样计数, 从而在尽可能降低存储开销的基础上, 实现高精度的每流测量. 基于真实网络数据集的仿真实验结果表明, 所提出的流大小估计器可以在片上高速存储占用仅为 22.75 kB 的情况下, 平均相对误差较已有机制降低接近 1 个数量级.

参考文献

1 Basat R B, Einziger G, Gong J Z, et al. Q-MAX: a unified scheme for improving network measurement throughput. In: Proceedings of the Internet Measurement Conference, 2019. 322–336

2 Shi S Q, Qian C, Wang M M. Re-designing compact-structure based forwarding for programmable networks. In: Proceedings of the 27th International Conference on Network Protocols (ICNP), 2019. 1–11

3 Shi S, Qian C. Ludo hashing: compact, fast, and dynamic key-value lookups for practical network systems. Proc ACM Meas Anal Comput Syst, 2020, 4: 1–32

4 Ahmed P, Hasan M, Kashyap A, et al. Efficient computation of top-k frequent terms over spatio-temporal ranges. In: Proceedings of the ACM International Conference on Management of Data, 2017. 1227–1241

5 Li Y, Zhang H, Cai Z P. An overview of software-defined network measurement technologies. Sci Sin Inform, 2018, 48: 293–314 [李阳, 张恒, 蔡志平. SDN 网络测量技术综述. 中国科学: 信息科学, 2018, 48: 293–314]

6 Cormode G, Muthukrishnan S. An improved data stream summary: the Count-Min sketch and its applications. J Algorithms, 2005, 55: 58–75

7 Lu Y, Montanari A, Prabhakar B, et al. Counter Braids: a novel counter architecture for per-flow measurement. SIGMETRICS Perform Eval Rev, 2008, 36: 121–132

8 Yang T, Zhang H, Li J, et al. HeavyKeeper: an accurate algorithm for finding top-k elephant flows. IEEE ACM Trans Netw, 2019, 27: 1845–1858

9 Li Y, Yu X, Yang Y, et al. Pyramid family: generic frameworks for accurate and fast flow size measurement. IEEE ACM Trans Netw, 2022, 30: 586–600

10 Sheng S, Huang Q, Wang S, et al. PR-Sketch: monitoring per-key aggregation of streaming data with nearly full accuracy. Proc VLDB Endow, 2021, 14: 1783–1796

11 Chen M, Chen S, Cai Z. Counter Tree: a scalable counter architecture for per-flow traffic measurement. IEEE ACM Trans Netw, 2017, 25: 1249–1262

12 Zhou Y, Zhou Y, Chen M, et al. Persistent spread measurement for big network data based on register intersection. Proc ACM Meas Anal Comput Syst, 2017, 1: 1–29

13 Homem N, Carvalho J P. Finding top-k elements in data streams. Inf Sci, 2010, 180: 4958–4974

14 Zhao B H, Li X, Tian B Y, et al. DHS: adaptive memory layout organization of sketch slots for fast and accurate data stream processing. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021. 2285–2293

15 Li J Z, Li Z K, Xu Y F, et al. WavingSketch: an unbiased and generic sketch for finding top-k items in data streams. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020. 1574–1584

16 Du Y, Huang H, Sun Y E, et al. Self-adaptive sampling for network traffic measurement. In: Proceedings of the IEEE Conference on Computer Communications, 2021. 1–10

https://engine.scichina.com/doi/10.1360/SSI-2023-0294

17 Einziger G, Fellman B, Friedman R, et al. ICE buckets: improved counter estimation for network measurement. IEEE ACM Trans Netw, 2018, 26: 1165–1178
18 Li H, Chen Q, Zhang Y, et al. Stingy Sketch: a sketch framework for accurate and fast frequency estimation. Proc VLDB Endow, 2022, 15: 1426–1438
19 Zhou Y, Yang T, Jiang J, et al. Cold filter: a meta-framework for faster and more accurate stream processing. In: Proceedings of the International Conference on Management of Data, 2018. 741–756
20 Jia P, Wang P H, Zhao J Z, et al. Loglog filter: filtering cold items within a large range over high speed data streams. In: Proceedings of the 37th International Conference on Data Engineering (ICDE), 2021. 804–815

Adaptive Sketch: accurate flow size measurement in high-speed networks

Xiaofei BU[1], He HUANG[2*], Yu-E SUN[3], Zhaojie WANG[2] & Xiaocan WU[2]

1. *College of Software, Shenyang Normal University, Shenyang* 110034*, China*;
2. *School of Computer Science and Technology, Soochow University, Suzhou* 215006*, China*;
3. *School of Rail Transportation, Soochow University, Suzhou* 215131*, China*
* Corresponding author. E-mail: huangh@suda.edu.cn

Abstract The measurement of flow size in high-speed networks faces a significant challenge due to the scarcity of high-speed memory resources, making it difficult to meet the real-time storage demands of massive flow data. Existing works commonly rely on memory-sharing techniques to place designed estimators in the limited high-speed memory. However, this approach introduces a substantial amount of noise that is hard to eliminate, leading to lower estimation accuracy for medium and small-scale flows. This paper proposes an Adaptive Sketch technique that adapts the memory space based on the flow size to address this issue. Building upon this technique, a high-precision, low-memory-cost flow size estimator is designed. The flow size estimator efficiently filters out massive noising/small flows using reversible counters and further employs sample counters with decreasing sampling probabilities at each level to adaptively sample different-sized flows. This technique effectively controls memory usage by large flows, achieving low cost and high precision in flow size measurement. Experiments based on the real network dataset CAIDA 2019 demonstrate that the proposed flow size estimator reduces the average relative error by nearly 1 order of magnitude compared to existing mechanisms.

Keywords high-speed network, flow size measurement, Sketch, small flows filtering, reversible counters

https://engine.scichina.com/doi/10.1360/SSI-2023-0294