

4 种计数型 Bloom Filter 的性能分析与比较^{*}

张进⁺, 邬江兴, 刘勤让

(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

Performance Evaluation and Comparison of Four Counting Bloom Filter Schemes

ZHANG Jin⁺, WU Jiang-Xing, LIU Qin-Rang

(National Digital Switching System Engineering & Technology Research Center, Zhengzhou 450002, China)

+ Corresponding author: E-mail: boost_zj@163.com

Zhang J, Wu JX, Liu QR. Performance evaluation and comparison of four counting bloom filter schemes. Journal of Software, 2010,21(5):1098–1114. <http://www.jos.org.cn/1000-9825/3495.htm>

Abstract: The Counting Bloom Filter (CBF) is a space-efficient data structure that extends a Bloom filter so as to allow approximate multiplicity queries on a dynamic multi-set. An in-depth study of three existing CBF schemes is presented, that is, the Naïve Counting Bloom Filter (NCBF), the Space-Code Bloom Filter (SCBF) and the d-left Counting Bloom Filter (dlCBF). Then, a CBF scheme called Binary Shrinking d-left Counting Bloom Filter (BSdlCBF) is proposed. A performance metrics named load adaptability for CBF schemes is also defined. The performance of the four CBF schemes is evaluated by using metrics of counting error, space complexity and load adaptability under both uniform and Zipfian multiplicity distributions. The experimental results show that the proposed BSdlCBF outperforms the other three in terms of accuracy, space-efficiency and load adaptability. The cost of such an advantage of BSdlCBF is a reasonable rise in computational and space complexity.

Key words: counting bloom filter; performance evaluation; performance comparison; load adaptability

摘要: 对 3 种已有的计数型 Bloom filter——Naïve Counting Bloom Filter(NCBF),Space-Code Bloom Filter(SCBF)和 d-left Counting Bloom Filter(dlCBF)——的查询错误概率进行了分析,得出了 NCBF 的计数器防溢出条件以及 SCBF 和 dlCBF 的参数最优设置准则.提出了一种衡量计数型 Bloom filter 性能的指标:负载适应性.针对 dlCBF 负载适应性差的问题,对 dlCBF 进行了改进,提出了一种计数型 Bloom filter:Binary Shrinking d-left Counting Bloom Filter(BSdlCBF).通过仿真实验,以计数误差、空间复杂度以及负载适应性为性能指标,对上述 4 种 CBF 进行了比较.实验结果表明,BSdlCBF 具有最低的空间复杂度、最小的计数误差以及最佳的负载适应性. BSdlCBF 赢得上述性能优势的代价在于其计算复杂度比其他 3 种计数型 Bloom filter 略高.

关键词: 计数型 Bloom filter;性能评估;性能比较;负载适应性

中图法分类号: TP393 文献标识码: A

Bloom filter 是一种支持集合的近似表示以及近似从属查询(approximate membership query)的数据结构^[1].

^{*} Supported by the National Basic Research Program of China under Grant Nos.2007CB307100, 2007CB307102 (国家重点基础研究发展计划(973))

Received 2008-06-18; Accepted 2008-10-27

设集合 S 是集合 U 的一个子集,对于 U 中的任一元素 e ,Bloom filter 可以一定的虚报概率判断 e 是否属于 S . Bloom Filter 突出优点是空间效率高:无论元素 e 自身的大小如何,每元素只需 10 比特,即可实现虚报概率约为 1%的近似从属查询.计数型 Bloom filter(counting bloom filter,简称 CBF)^[2-6]将 Bloom filter 的从属查询扩展为更一般的频率查询(multiplicity query,或者称为计数查询),允许 S 为动态多集,这极大地拓展了 Bloom filter 的应用领域.CBF 在网络、数据库等众多领域得到了广泛的应用^[3,4,6-9].

文献[2]提出了 Naïve Counting Bloom Filter(NCBF),文献[7]采用 NCBF 进行骨干网链路的长流流量测量;文献[4]提出了 Space-Code Bloom Filter(SCBF)用于高速骨干网的逐流流量测量;文献[6]提出了 d-left Counting Bloom Filter(dICBF),文献[9]采用 dICBF 构建大规模并发状态机.上述 3 种 CBF 结构简单,易于硬件实现,特别适合于高速数据的实时分析处理.除了上述 3 种 CBF 之外,还有文献[3,5]提出的两种 CBF 结构.但是,文献[3,5]中提出的 CBF 比上述 3 种 CBF 复杂,难以硬件实现,因此其应用范围也比上述 3 种 CBF 狭窄^[6],故本文不将这两种 CBF 作为讨论的对象.

由于各种不同的 CBF 在计算复杂度、空间复杂度以及计数误差等不同的性能指标之间有着不同的权衡,因此在实际使用中应当根据应用需求和资源约束,选择最合适的一种 CBF.为了给 CBF 的设计选择提供准确而详尽的参考依据,对各种 CBF 进行系统、定量的分析比较显得尤为必要.然而,已有的研究工作中,比较的对象以及比较所采用的性能指标都不完备(已有的相关工作详见第 6 节).针对此问题,本文采用多种性能评价尺度,从不同角度对 NCBF,SCBF,dICBF 以及本文提出的 BSdICBF 进行了横向比较.具体而言,本文的主要贡献如下:

(1) 对已有的 3 种 CBF,即 NCBF,SCBF 以及 dICBF 进行深入分析,给出了 NCBF 的防溢出条件以及 SCBF 和 dICBF 的参数最优设置准则,为上述 CBF 在实际使用时的参数最优设计提供了理论依据.

(2) 提出了一种新的衡量 CBF 性能的指标:负载适应性.本文分析比较的 4 种计数型 Bloom filter 由于结构简单,因而常用于高速数据处理场合,如高速骨干网流量分析^[4,7,9].高速数据处理应用中,算法一般采用硬件实现.在硬件实现中,计算与存储资源往往根据系统的最大负载预先设定.然而在实际应用中,系统负载通常是不稳定的,并且在大部分时间中,负载都远小于最大值.例如现有研究表明,在骨干网络链路中,最大并发数据流的数量会达到 $1M$ ^[3,6].然而,对实际采集的高速骨干网流量数据的分析表明,绝大部分情况下,链路上的并发流的数量都远小于最大值^[10].本文将计数型 Bloom filter 在低负载时,充分利用已分配资源,降低计数误差的能力称为计数型 Bloom filter 的负载适应性.对于硬件实现而言,算法的负载适应性是一项重要的性能指标.

(3) 提出了一种新的计数型 Bloom filter:BSdICBF.我们发现,已有的 3 种 CBF(即 NCBF,SCBF 和 dICBF)中,虽然从计数误差、空间复杂度的角度而言 dICBF 的性能最佳,但是,dICBF 的负载适应性却明显比 NCBF 要差.针对 dICBF 负载适应性差的问题,本文对 dICBF 进行了改进,提出了一种计数型 Bloom filter:BSdICBF. BSdICBF 采用折半收缩机制,根据负载轻重程度动态地调整元素指纹的长度.在相同的空间复杂度下,满负载时,BSdICBF 和 dICBF 的性能一致;轻负载时,分析和实验的结果表明,BSdICBF 的计数误差明显小于 dICBF.分析结果表明,BSdICBF 的计算复杂度和空间复杂度略高于 dICBF.

(4) 通过仿真实验的手段,采用计数误差、空间复杂度以及负载适应性为性能指标(这些指标的定义将在第 1 节给出),分别从 3 个不同的角度,对上述 4 种 CBF 的性能进行了系统的比较.首先,本文比较了相同空间复杂度下,NCBF,SCBF 和 dICBF 的计数误差;其次,本文比较了在计数误差相近的前提下,NCBF,SCBF 和 dICBF 的空间复杂度.上述两项比较工作均在满负载的条件下进行,由于在满负载时,BSdICBF 和 dICBF 的性能一致,故上述两项性能比较无须包含 BSdICBF;最后,本文比较了 4 种 CBF 的负载适应性.仿真实验结果表明:1) 在满负载且空间复杂度相同时,SCBF 的计数误差最大,NCBF 次之,dICBF 的计数误差比 NCBF 和 SCBF 小若干个数量级;2) 在满负载且计数误差相近时,SCBF 的空间复杂度最高,NCBF 次之,dICBF 的空间复杂度最低;3) BSdICBF 具有最好的负载适应性,NCBF 次之,dICBF 的负载适应性比 NCBF 差,SCBF 的负载适应性最差.综上,在本文讨论的 4 种 CBF 中,BSdICBF 性能最佳.BSdICBF 赢得性能优势的代价是其计算复杂度比其他 3 种 CBF 略高.

本文第 1 节定义衡量 CBF 性能的指标.第 2 节对已有的 3 种 CBF 的计数误差进行深入分析,给出 NCBF 的防溢出条件以及 SCBF 和 dICBF 的参数最优设置准则.第 3 节提出一种计数型 Bloom filter:BSdICBF,并分析其

错误概率、计算复杂度和空间复杂度.第4节阐述并比较4种CBF性能时所采用的方法.第5节给出实验结果与分析.第6节介绍相关的工作.第7节为全文总结.

1 性能指标

对于计数查询而言,不仅仅需要关注错误概率,还需关注在错误发生时相对误差的大小.错误概率和相对误差的定义如下:

设 U 为全体元素构成的集合, $S(|S|=N)$ 为多集(multi-set),且 $S \subseteq U$.对于任意 $e \in U$,设 $f_e(f_e \in [0, F])$ 为 e 在 S 中出现的频率(或次数), $f_e=0$ 意味着 $e \notin S$. \hat{f}_e 为 f_e 的查询估计值.

定义 1(错误概率). $\forall e \in U, i \in [0, F]$, 定义频率为 i 的元素的错误概率为元素频率的估计值不等于真实值的概率.错误概率记作 $EP_i = P(\hat{f}_e \neq i)$. 称 $\hat{f}_e > f_e$ 为阳性错误, $\hat{f}_e < f_e$ 为阴性错误.

需要指出的是,本文所讨论的4种CBF中,只有SCBF的错误概率与元素频率是相关的,其他3种CBF的错误概率与元素频率无关.为了不失一般性,本文将错误概率定义为与元素频率相关的形式.当错误概率与元素频率无关时, EP_i 也可记作 EP .

定义 2(相对误差). $\forall e \in S, i \in [1, F]$, 定义频率为 i 的元素的相对误差为 S 中所有频率为 i , 并且发生查询错误(即 $\hat{f}_e \neq f_e$) 的元素的相对误差的期望.相对误差记作 $RE_i = E\left(\frac{|\hat{f}_e - i|}{i}\right), \hat{f}_e \neq f_e = i$.

CBF的计数精度用计数误差来衡量.计数误差的定义如下:

定义 3(计数误差). $\forall e \in S, i \in [1, F]$, 定义 $CE_i = EP_i \cdot RE_i$ 为频率为 i 的元素的计数误差.

本文采用不同负载率下的归一化错误概率和归一化计数误差来衡量4种CBF的负载适应性.归一化错误概率和归一化计数误差的定义如下:

定义 4(归一化错误概率). 定义负载率为 $\rho(0 < \rho \leq 1)$ 时的错误概率和满负载时的错误概率之比为归一化错误概率.归一化错误概率记作 $NEP_i(\rho) = EP_i(\rho) / EP_i(1)$.

定义 5(归一化计数误差). $\forall i \in [1, F]$, 定义负载率为 $\rho(0 < \rho \leq 1)$ 时的计数误差和满负载时的计数误差之比为归一化计数误差.归一化计数误差记作 $NCE_i(\rho) = CE_i(\rho) / CE_i(1)$.

图1为应用Bloom filter进行数据处理的一般系统结构模型,新到达的数据可能触发更新操作或查询操作.更新操作指的是对于新到达的元素 e ,更新其在计数型Bloom filter中的频率计数值;查询操作指的是读取元素 e 在计数型Bloom filter中的频率计数值,并据此给出其频率的估计值 \hat{f}_e .需要指出的是,某些计数型Bloom filter(如SCBF)的更新操作不需要对存储单元进行读操作,而仅仅进行写操作.但为了不失一般性,如图1所示的更新操作包括读存储单元操作和写存储单元操作.在处理海量高速数据的应用场合(如高速骨干网流量分析),处理单元通常由ASIC, FPGA或者专用CPU等高性能处理器件实现;存储单元通常由高速SRAM实现.此时,系统的性能瓶颈在于存储单元的带宽和容量,这两项性能指标分别用计算复杂度和空间复杂度来衡量^[4].计算复杂度定义为每次更新或者查询操作中处理单元访问存储单元的次數;空间复杂度定义为存储单元的空间大小^[4].

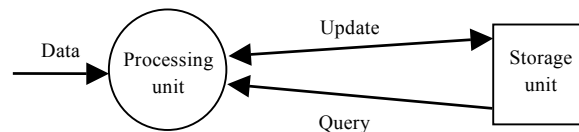


Fig.1 A general system model of data processing using Bloom filters

图1 应用 Bloom filter 进行数据处理的一般系统结构模型

2 3种已有CBF的计数误差分析

在具体分析每种CBF的计数误差之前,首先说明一下本文的符号使用约定.为了便于区分,对于每种CBF,

本文采用其缩写的首字母作为其特定参数的下标,例如 NCBF 的参数记作 k_n, m_n, w_n , SCBF 的参数记作 g_s, b_s 等等(上述参数的具体含义在下文分析中会详细说明).

2.1 NCBF 的计数误差分析

NCBF 由 k_n 个独立的哈希函数 $h_1(\cdot), h_2(\cdot), \dots, h_{k_n}(\cdot)$ 和一个计数器向量 C 构成. 设 C 中共包含了 m_n 个计数器, 每个计数器的宽度为 w_n 比特. 当更新元素 e 的计数时, 将计数器 $C[h_1(e)], C[h_2(e)], \dots, C[h_{k_n}(e)]$ 分别增加 1. 当查询 e 的频率 f_e 时, 取 $C[h_1(e)], C[h_2(e)], \dots, C[h_{k_n}(e)]$ 中最小的作为 f_e 的估计值, 即 $\hat{f}_e = \arg \min_i (C[h_i(e)])$, $i \in [1, k_n]$.

NCBF 的错误概率即为 Bloom filter 的虚报概率^[3], 即

$$EP_n = \left(1 - \left(1 - \frac{1}{m_n} \right)^{k_n N} \right)^{k_n} \approx \left(\frac{1}{2} \right)^{k_n} \quad (1)$$

其中, $|N|=S, k_n = \ln(2) \cdot m_n / N$.

文献[3,7]提出, 采用最小增加(minimum increase, 简称 MI)策略可以有效地降低 NCBF 的计数误差. 当更新元素 e 的计数时, 最小增加策略仅仅将计数器 $C[h_1(e)], C[h_2(e)], \dots, C[h_{k_n}(e)]$ 中最小的计数器增加 1, 其他计数器的值设置为 $\arg \min_i (C[h_i(e)]) + 1$ 与其原先的值中较大的那个. 在采用 MI 策略时, NCBF 的错误概率将难以分析. 第 5 节通过仿真实验得出了采用 MI 策略时 NCBF 的计数误差.

由于且仅由于 NCBF 计数器的计数范围限制, 会导致 NCBF 产生阴性错误($\hat{f}_e < f_e$). 定理 1 给出了与阳性错误相比, NCBF 的阴性错误可以忽略不计的条件. 定理 1 的证明见附录 A.

定理 1. 当 CBF 计数器的计数上限为 $2F$ 时, 即当

$$w_n = \lceil \log_2(2F+1) \rceil \quad (2)$$

时, NCBF 发生阴性错误的概率与发生阳性错误的概率之比小于 0.07^{k_n} .

NCBF 的主要参数见表 1. 在实验中, 我们根据公式(2)设置 w_n 的取值, m_n 在一定的范围内变化. 实验细节见第 4 节和第 5 节. 结论 1 给出了 NCBF 的计算复杂度和空间复杂度.

Table 1 Dominant parameters of NCBF

表 1 NCBF 的关键参数

Parameter	Description
m_n	Length of counter vector
w_n	Counter width, set according to Formula (2)

结论 1. NCBF 更新操作的计算复杂度为 $2k_n$. NCBF 查询操作的计算复杂度为 k_n . NCBF 的空间复杂度为 $m_n \cdot w_n$.

2.2 SCBF 的计数误差分析

Kumar 等人提出了 Space-Code Bloom Filter(SCBF)用于高速骨干网的流量测量^[4]. 如果将 NCBF 采用的 k_n 个哈希函数看作一个哈希函数组 H , 则 SCBF 由 g_s 个哈希函数组 $H_1 = \{h_1^1(\cdot), h_2^1(\cdot), \dots, h_{k_n}^1(\cdot)\}, H_2 = \{h_1^2(\cdot), h_2^2(\cdot), \dots, h_{k_n}^2(\cdot)\}, \dots, H_{g_s} = \{h_1^{g_s}(\cdot), h_2^{g_s}(\cdot), \dots, h_{k_n}^{g_s}(\cdot)\}$ 和一个比特向量 B 构成, B 的长度为 b_s 比特. 当更新元素 e 的计数时, 从 g_s 个哈希函数组中随机地选择一组(不妨设为 H_i)对 e 作哈希运算, 然后将比特向量 B 的对应位 $B(h_1^i(e)), B(h_2^i(e)), \dots, B(h_{k_n}^i(e))$ 置为 1. 对于元素 e 而言, 每次更新时都可能选择不同的哈希函数组. 当估计 e 的出现频率 f_e 时, 则遍历查询所有的哈希函数组. 对于哈希函数组 i , 若 $B(h_1^i(e)), B(h_2^i(e)), \dots, B(h_{k_n}^i(e))$ 均为 1, 则称该哈希函数组与元素 e 匹配. 设遍历查询 g_s 个哈希函数组后, 发现与元素 e 匹配的哈希函数组的数量为 θ . 然后, 利用某种估计算法根据 θ 估算出 f_e , 即 $\hat{f}_e = estimator(\theta)$. 文献[4]提出了最大似然估计和最小均方误差估计这两种估计器. SCBF 每组哈希函数的数量为

$$k_s = \ln(2) \cdot \frac{b_s}{N \cdot E(f)} \quad (3)$$

其中, $E(f)$ 为元素频率的期望值.

SCBF 由于采用了估计过程,其查询错误概率不再与元素的出现频率无关,因而难以分析.同时,估计器还会引起阴性错误.

对参数 g_s 的最优设置问题需要进行深入的讨论.文献[4]指出,若 g_s 选取得过小,则估计器会存在较大的估计误差;若 g_s 选取得过大,则会由于 Bloom filter 阳性错误的影响,使得观测值 θ 中噪声成分增加,从而引起较大的观测误差.但是,文献[4]并未给出 g_s 的最优设置准则.本文通过实验给出了 g_s 的最优设置准则,方法如下:令 k_s 分别取 1 到 10,在某一 k_s 取值下,设置 g_s 分别为 $1F \sim 10F$,然后分别计算各个 g_s 取值下 SCBF 的平均计数误差.在某一 k_s 取值下,导致平均计数误差最小的 g_s 取值为当前 k_s 取值下 g_s 的最佳值.表 2 给出了不同的 k_s 值下 g_s 的最佳值的计算结果.

Table 2 Optimal g_s under various k_s

表 2 不同的 k_s 值下 g_s 的最优取值

		k_s									
		1	2	3	4	5	6	7	8	9	10
Estimator	MLE	1F	1F	1F	1F	1F	2F	2F	3F	4F	8F
	MVE	1F	1F	1F	1F	1F	2F	2F	3F	5F	7F

表 3 列出了 SCBF 的重要参数.结论 2 给出了 SCBF 的计算复杂度和空间复杂度.

Table 3 Dominant parameters of SCBF

表 3 SCBF 的关键参数

Parameter	Description
b_s	Length of bit vector
g_s	Number of hash groups

结论 2. SCBF 更新操作的计算复杂度为 k_s . SCBF 查询操作的计算复杂度为 $k_s \cdot g_s$. SCBF 的空间复杂度为 b_s .

2.3 dICBF 的计数误差分析

Bonomi 等人提出了一种采用 d-left 哈希函数结合元素指纹来构建计数型 Bloom filter 的方法(d-left counting bloom filter,简称 dICBF)^[6],如图 2 所示.d-left 哈希函数^[11]将存储区等分为 d 个块,每块又分为若干个相同容量的桶(d 个块的空间大小可以不相等.本文为了论述方便,假设 d 个块的空间相同).不妨将各个块看作桶向量(bucket vector),从左向右依次记作 BV_1, BV_2, \dots, BV_d .例如,图 2 中的 d-left 哈希函数的存储区划分为 4 个块,每块 5 个桶,每桶深度为 4.当插入元素 e 时,由 d 个独立的哈希函数计算元素 e 在各个块中的桶地址,分别记作 $h_1(e), h_2(e), \dots, h_d(e)$.然后将 e 插入到 $BV_1(h_1(e)), BV_2(h_2(e)), \dots, BV_d(h_d(e))$ 中负载最轻的那个桶中.如果存在多个负载最轻的桶,则选择最左边那个.例如,图 2 是将元素 e 插入到桶 $BV_1(4)$ 中.遵循上述选择策略,可使得各个桶的负载较为平均,从而各个桶在平均负载的基础上再增加一个较小的额外桶空间,即可保证桶的溢出概率极低,从而有效地提高空间利用率^[11].当额外桶空间一定时,d-left 哈希函数的并行选择机会越多(即 d 值越大),桶溢出概率越低(对桶溢出概率的分析参见附录 B).

设元素指纹的长度为 l_d^f 比特,计数器位宽为 l_d^m ($l_d^m = \lceil \log_2(F+1) \rceil$) 比特,由于 d-left 哈希函数是一种近乎完美的哈希函数,因此, BV 的空间略大于 $|S| \cdot (l_d^f + l_d^m)$ 比特.设每桶的平均深度为 b_d^m ,额外深度为 b_d^e .由文献[12]可知,取 $d=4$ 及 $b_d^e=1$ 即可保证极低的桶溢出概率.在本文中,为了便于分析, d 和 b_d^e 的取值均遵循此设置**.由文献[6]可知,dICBF 的错误概率为

** 下文的分析均基于这一参数设置.虽然在此设置下得出的结论不具有通用性,但是我们的分析方法仍然具有普适性和一般性.

$$EP_d = 1 - \left(1 - \frac{1}{2^{l_d^f}}\right)^{b_d^m \cdot d} \quad (4)$$

当 l_d^f 较大,如 $l_d^f \geq 8$ 时,上式近似化简为^[6]

$$EP_d \approx \frac{b_d^m \cdot d}{2^{l_d^f}} = \frac{4b_d^m}{2^{l_d^f}} \quad (5)$$

dICBF 的每个元素所需空间 l_d 为

$$l_d = \frac{b_d^m + b_d^e}{b_d^m} \cdot (l_d^f + l_d^m) = \frac{b_d^m + 1}{b_d^m} \cdot (l_d^f + \lceil \log_2(F + 1) \rceil) \quad (6)$$

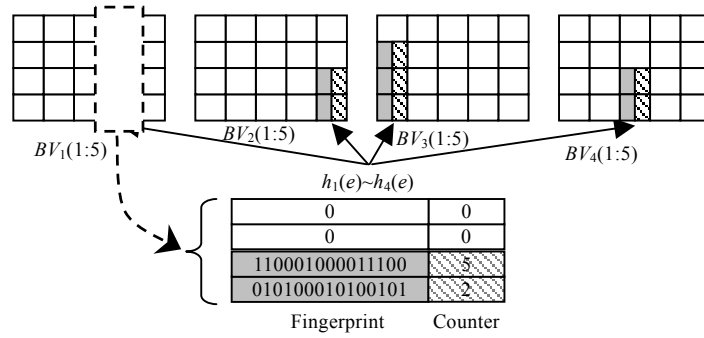


Fig.2 Structure illustration of dICBF

图 2 dICBF 的结构示意图

通过分析 dICBF 的每个元素所需空间 l_d 和 dICBF 错误概率 EP_d 的关系我们发现,存在着最优的 l_d^f 和 b_d^m 的取值,使得在 l_d 一定的前提下,错误概率 EP_d 最小.定理 2 给出了 l_d^f 和 b_d^m 的最优取值.

定理 2. 当每元素空间 l_d 一定时, l_d^f 和 b_d^m 如下取值可使得 dICBF 的错误概率 EP_d 最小:

$$l_d^f = \left\lceil \frac{1}{2} \cdot \left(l_d - 2l_d^m + \sqrt{l_d^2 - \frac{4l_d}{\ln(2)}} \right) \right\rceil \quad (7)$$

且

$$b_d^m = \left\lceil \left(l_d + \sqrt{l_d^2 - \frac{4l_d}{\ln(2)}} \right) / \left(l_d - \sqrt{l_d^2 - \frac{4l_d}{\ln(2)}} \right) \right\rceil \quad (8)$$

定理 2 的证明见附录 C.表 4 列出了 dICBF 的重要参数. l_d^f 和 b_d^m 的取值根据公式(7)和公式(8)设置.结论 3 给出了 dICBF 的计算复杂度和空间复杂度.

Table 4 Dominant parameters of dICBF

表 4 dICBF 的关键参数

Parameter	Description
d	Number of parallel hash in d-left, set $d=4$
b_d^e	Extra bucket depth, set $b_d^e=1$
l_d	Per-Element space

结论 3. dICBF 更新操作的计算复杂度为 $d \cdot (b_d^m + b_d^e) + 1$. dICBF 查询操作的计算复杂度为 $d \cdot (b_d^m + b_d^e)$. dICBF 的空间复杂度为 $l_d \cdot N$,其中, N 为集合 S 中元素的个数, l_d 由公式(6)给出.

3 改进的 dICBF:BSdICBF

3.1 BSdICBF的结构描述

BSdICBF 根据桶负载大小,采用折半收缩的机制,动态地调整元素指纹的长度.图 3 给出了当 dICBF 的桶空间为 8,桶负载分别为 1,2,4,8 时,BSdICBF 和 dICBF 桶空间利用情况对比(阴影方格表示元素指纹,斜线方格标识频率计数器).由图 3 可以看出,当轻负载时,BSdICBF 能够充分地利用桶空间,扩充元素指纹.因此在轻负载条件下,采用折半收缩机制可以有效的降低错误概率.

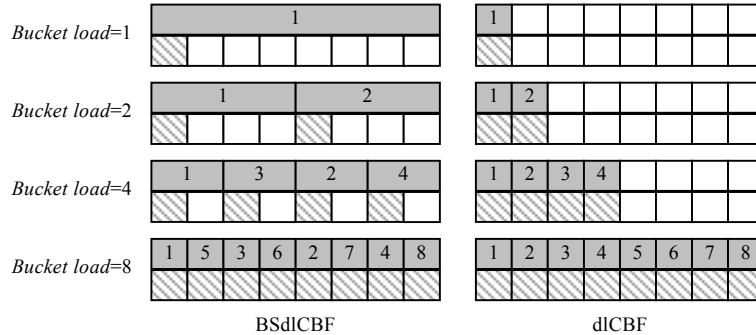


Fig.3 Comparison of bucket space occupation of BSdICBF and dICBF under various load rate

图 3 不同负载率下,BSdICBF 和 dICBF 的桶空间利用情况对比

令 $l = l_d^f$, 称 l 为单位指纹长度;令 $b = b_a^m + b_a^s$.不妨设 b 为 2 的自然数次幂(即 $b=2^x, x \in \mathbb{N}$).对于每个元素 e , BSdICBF 生成 b 个 l 比特的元素指纹 $F_1(e), F_2(e), \dots, F_b(e)$.设当向桶中插入元素 e 时,桶中已有 $i-1$ 个元素,则将 $F_1(e), F_2(e), \dots, F_{L(i)}(e)$ 写入桶中从 $A(i)$ 开始的 $L(i)$ 个单元即可, $L(i)$ 和 $A(i)$ 分别由公式(9)和公式(10)确定.

$$L(i) = \frac{b}{2^{\lceil \log_2 i \rceil}} \tag{9}$$

$$A(i) = 1 + (2 \cdot (i - 2^{\lceil \log_2 i \rceil - 1}) - 1) \cdot L(i) \tag{10}$$

$L(i)$ 的单位为单位指纹长度, $A(i)$ 为指纹单元地址.由公式(2)可知,当 $i > b/2$ 时 $L(i)=1$, 此时, BSdICBF 退化为元素指纹长度为 1 单位指纹长度的 dICBF.当 b 不等于 2 的自然数次幂时,我们作如下处理:设 $2^j < b < 2^{j+1}$, 当向桶中插入第 i 个元素时,其元素地址 $A(i)$ 和指纹长度 $L(i)$ 分别为

$$L(i) = \begin{cases} 2^{j - \lceil \log_2 i \rceil}, & i \leq 2^j \\ 1, & 2^j < i \leq b \end{cases} \tag{11}$$

$$A(i) = \begin{cases} A(i) = 1 + (2 \times (i - 2^{\lceil \log_2 i \rceil - 1}) - 1) \cdot L(i), & i \leq 2^j \\ i, & 2^j < i \leq b \end{cases} \tag{12}$$

BSdICBF 的缺陷在于,由于使用了折半收缩机制,使其无法支持删除操作.另外,BSdICBF 的计算复杂度和空间复杂度比 dICBF 略高.关于 BSdICBF 复杂度的分析详见下文.

3.2 BSdICBF的错误概率分析

设共向 BSdICBF 中插入了 x 个元素,则称 $r = \frac{x}{\sum_{i=1}^d |BV_i|}$ 为平均桶负载.定理 3 给出了在任意平均桶负载条件下,BSdICBF 的错误概率.

定理 3. 设单位指纹长度为 l , 桶深为 b , 则当平均桶负载为 r 且没有发生桶溢出时,BSdICBF 的错误概率 EP_{Bd} 为

定理 3. 设单位指纹长度为 l , 桶深为 b , 则当平均桶负载为 r 且没有发生桶溢出时,BSdICBF 的错误概率 EP_{Bd} 为

$$EP_{Bd} = 1 - \left(P(r, 0) + \sum_{i=1}^b P(r, i) \left(1 - \frac{1}{2^{L(i)-l}} \right)^i \right)^d \quad (13)$$

其中, $P(r, i)$ 为 d -left 哈希函数的平均桶负载为 r 时, 对于任意一个桶, 其中有 i 个元素的概率 ($P(r, i)$ 的计算方法见附录 B), $L(i)$ 由公式(11)确定.

证明: BSdICBF 的错误概率为对于任意一个元素指纹 fp , 在任意 d 个桶中, 至少存在 1 个与之相同的指纹的概率. 设 EP' 为对于任意一个桶, 其中的元素指纹都与 fp 不一致的概率, 则有:

$$EP_{Bd} = 1 - (EP')^d \quad (14)$$

设 EP'_i 为当桶中有 $i (1 \leq i \leq b)$ 个元素, 并且这 i 个元素指纹都与 fp 不一致的概率, 则由全概率公式, 有:

$$EP' = P(r, 0) + P(r, 1) \cdot EP'_1 + P(r, 2) \cdot EP'_2 + \dots + P(r, b) \cdot EP'_b \quad (15)$$

考虑到

$$EP'_i = \left(1 - \frac{1}{2^{L(i)-l}} \right)^i \quad (16)$$

于是, 将公式(16)带入公式(15), 有:

$$EP' = P(r, 0) + P(r, 1) \left(1 - \frac{1}{2^{L(1)-l}} \right) + P(r, 2) \left(1 - \frac{1}{2^{L(2)-l}} \right)^2 + \dots + P(r, b) \left(1 - \frac{1}{2^{L(b)-l}} \right)^b \quad (17)$$

将公式(17)带入公式(14), 定理 3 得证. □

定理 3 成立的条件是桶溢出概率为 0. 在实际应用中, 为了保证桶溢出概率较低, 一般设置桶深大于期望的平均桶负载. 例如, 当 $d=4, b-r=1$ 时, 桶溢出概率为 10^{-10} 数量级; 当 $d=4, b-r=2$ 时, 桶溢出概率为 10^{-140} 数量级, 可以忽略不计. 可见, 在实际使用中, 定理 3 已经能足够精确地描述 BSdICBF 的错误概率. 表 5 为 $b_d^m = 3, b_d^e = 1, d=4$, l 分别取 4 和 8 时, BSdICBF 的错误概率的理论值和实验值的比较. 实验中, 元素数量 $N=10^6$, 平均桶负载以 0.5 为步长, 从 3 减小到 0.5. 由表 5 可以看出, 定理 3 给出的 BSdICBF 错误概率的理论值和实验结果较为接近.

Table 5 Comparison of theoretical and experimental error probability of BSdICBF

表 5 BSdICBF 错误概率的理论值与实验值比较

		$r=3$	$r=2.5$	$r=2$	$r=1.5$	$R=1$	$r=0.5$
$l=4$	Experimental value	0.485 4	0.321 5	0.107 6	0.015 9	0.003 9	6.0000e-5
	Theoretical value	0.505 7	0.329 1	0.108 6	0.017 1	0.003 9	6.1264e-5
$l=8$	Experimental value	0.039 7	0.022 8	0.006 2	1.4200e-4	1.2000e-5	2.2700e-7
	Theoretical value	0.042 1	0.024 1	0.006 0	1.3941e-4	1.1165e-5	2.6870e-7

3.3 BSdICBF 的复杂度分析

与 dICBF 相比, BSdICBF 在计算复杂度和空间复杂度上均有所提高. 定理 4 和定理 5 分别给出了 BSdICBF 和 dICBF 的空间复杂度与时间复杂度之比.

定理 4. BSdICBF 和 dICBF 的空间复杂度之比为 $1 + \frac{\lceil \log_2(b+1) \rceil}{b(l_d^f + l_d^m)}$, 其中, l_d^f 和 l_d^m 分别为 dICBF 的元素指纹长度和元素频率计数器位宽, b 为 dICBF 的桶深.

证明: 由于 BSdICBF 的每个桶需要设置一个负载计数器, 以记录当前桶中的元素数量, 因此, BSdICBF 比 dICBF 多需要 $\lceil \log_2(b+1) \rceil \sum_{i=1}^d |BV_i|$ 比特的存储空间. 考虑到 dICBF 所需的存储空间为 $b(l_d^f + l_d^m) \sum_{i=1}^d |BV_i|$ 比特, 于是, BSdICBF 和 dICBF 的空间复杂度之比为

$$1 + \frac{\lceil \log_2(b+1) \rceil \sum_{i=1}^d |BV_i|}{b(l_d^f + l_d^m) \sum_{i=1}^d |BV_i|} = 1 + \frac{\lceil \log_2(b+1) \rceil}{b(l_d^f + l_d^m)}$$

定理 4 得证. □

定理 5. BSdICBF 和 dICBF 的查询操作的计算复杂度之比为 1,BSdICBF 和 dICBF 的更新操作的计算复杂度之比小于 $1+1/d$.

证明:设平均桶负载为 r ,则 BSdICBF 每次查询操作均需要访问 $b \cdot d$ 个桶单元,而每次更新操作需要访问 $b \cdot d + \log_2(L(\lceil r \rceil))$ 个桶单元.考虑到 dICBF 查询操作的计算复杂度为 $b \cdot d$,更新操作的计算复杂度为 $b \cdot d + 1$,于是,BSdICBF 和 dICBF 的查询操作的计算复杂度之比为 1,BSdICBF 和 dICBF 的更新操作的计算复杂度之比为

$$\frac{b \cdot d + \log_2(L(\lceil r \rceil))}{b \cdot d + 1} \leq \frac{b \cdot d + b}{b \cdot d + 1} = 1 + \frac{b-1}{b \cdot d + 1} < 1 + \frac{b-1}{b \cdot d - d} = 1 + \frac{1}{d}.$$

定理 5 得证. □

由定理 5 和定理 6 可以看出,与 dICBF 相比,BSdICBF 的计算复杂度和空间复杂度略有提高.此外,有两个原因使得 BSdICBF 的处理单元的处理代价也略有提高:首先,BSdICBF 的哈希运算所产生的元素指纹长度为 dICBF 的 b 倍,其中 b 为 dICBF 的桶深;其次,在进行元素指纹匹配时,dICBF 为定长指纹匹配,而 BSdICBF 为变长指纹匹配.然而,由于哈希运算可以并行进行,并且实际应用场合中往往有实现代价较低的、几乎理想的哈希函数可供选择^[13],因而提高的哈希运算代价并非不可接受.假设对于某一元素 e ,在 dICBF 中的元素指纹为 f ,在 BSdICBF 中的元素指纹为 $F[1:b](F[i](1 \leq i \leq b))$ 的长度和 f 的长度一致.对于 e 所可能插入的 d 个桶中的任意一个,设其中的元素指纹为 $B[1:b]$.在 dICBF 中查找元素 e 时,只需将 f 分别与 $B[i](1 \leq i \leq b)$ 进行比较即可.在 BSdICBF 中查找元素 e 时,设当前桶负载为 $r(1 \leq r \leq b)$,则只需将 $F[\text{map}(r,i)]$ 与 $B[i]$ 进行比较即可,其中, $\text{map}(r,i)$ 为总大小为 $b \cdot b$ 的映射表,例如 $\text{map}(1,i)=i, \text{map}(b,i)=1$,等等. $\text{map}(r,i)$ 可预先计算好,存放在片内的高速寄存器中.可见,变长指纹匹配的代价同样也可以接受.

综上所述,与 dICBF 相比,BSdICBF 的计算复杂度和空间复杂度并没有明显的提高.

4 比较方法

本文从 3 个不同的角度分别对上述 4 种 CBF 的性能进行了比较:首先,本文比较了在相同的空间复杂度下 4 种 CBF 的计数误差;其次,本文比较了在相近的计数误差的条件下 4 种 CBF 的空间复杂度;最后,本文对 4 种 CBF 的负载适应性进行了比较.前两项比较工作在满负载时进行.由于在满负载时,BSdICBF 的性能与 dICBF 一致,因此前两项比较工作仅需要包括 NCBF,SCBF 和 dICBF 这 3 种 CBF 即可.

为了比较 NCBF,SCBF 和 dICBF 的计数误差,首先需要设置 3 种 CBF 的空间复杂度相等.假设 $\eta = m_n/N$,则对于 NCBF 而言,每元素所需空间为 $\eta \cdot w_n$ 比特.于是,我们令 $l_d = \eta \cdot w_n$ 且 $b_s = m_n \cdot w_n$,则可以保证 dICBF 和 SCBF 的空间复杂度与 NCBF 相同.实验中我们保持 N 不变,在一定范围内改变 m_n ,分别比较不同的 η 取值下 3 种 CBF 的计数误差.

为了比较 NCBF,SCBF 和 dICBF 的空间复杂度,首先需要设置 3 种 CBF 的计数误差相等.由于无法保证 3 种 CBF 的计数误差严格相等,因此我们控制 3 种 CBF 的平均计数误差之差的绝对值小于某一接近于 0 的常数 ε ,使之计数误差相近.对于某一固定的 η ,为使 dICBF 和 NCBF 的计数误差相近, l_d 首先取一个较小的值,然后每次增加 1,直到两者计数误差之差的绝对值小于 ε .同样地,对于某一 η ,为使 SCBF 和 NCBF 的计数误差相近,首先设置 b_s 为 $b_s^0 = w_n \cdot m_n$,然后每次增加 Δb_s ,其中,

$$\Delta b_s = \frac{N \cdot E(f)}{\ln(2)} \quad (18)$$

直到两者计数误差之差的绝对值小于 ε .选择每次增加 Δb_s 的原因是, b_s 每增加 Δb_s , k_s 恰好增加 1.这样便于计算阳性错误概率以及根据表 2 确定 g_s 的取值.

在比较 4 种 CBF 的负载适应性时,首先采用进行 CBF 空间复杂度比较时的参数设置,通过适当的参数设置,使得 4 种 CBF 的满负载计数误差相近;然后,再逐步降低负载率,比较不同负载率下,4 种 CBF 的归一化错误概率和归一化计数误差.

5 实验结果与分析

实验中我们设置 $F=15, N=4096$. 元素的出现频率在区间 $[1, F]$ 内分别服从均匀分布和参数为 1.5 的 Zipf 分布. 之所以选择 Zipf 分布, 是因为考虑到在诸多典型的 CBF 应用场合, 如网络流量测量与分析中, 实际的元素频率较好地服从 Zipf 分布^[4,7,8]. 第 5.1 节比较满负载时, 在相同空间复杂度下, NCBF, SCBF 和 dICBF 的计数误差. 第 5.2 节比较满负载时, 在计数误差相近的条件下, NCBF, SCBF 和 dICBF 的空间复杂度. 在第 5.3 节中, 我们将对 4 种 CBF 的负载适应性进行比较.

5.1 计数误差比较

图 4 和图 5 分别为元素频率服从均匀分布和 Zipf 分布时, 相同空间复杂度下, 3 种 CBF 的计数误差比较. 从图 4 和图 5 可以看出, dICBF 的计数精度最高, NCBF 次之, SCBF 最差. 另外, 从图 4(f) 和图 5(f) 还可以看出, 当 $\eta=4$ 时, dICBF 的计数误差比 NCBF 和 SCBF 小若干个数量级. 随着 η 的增大, 这一差异更加明显. 当 $\eta=5$ 或 $\eta=6$ 时, dICBF 在某些元素频率点的计数误差为 0, 因而无法在对数坐标上画出. 因此, 本文没有给出图 4(c)、图 4(d)、图 5(c) 以及图 5(d) 的对数坐标图.

比较图 4 和图 5 可以发现, 在相同的 η 值下, Zipf 分布时的计数误差要比均匀分布时的计数误差小. 对于 NCBF 和 dICBF 而言, 虽然错误概率不受元素频率分布的影响, 但是相对误差却与元素频率分布相关. Zipf 分布下的元素频率的期望比均匀分布下的期望要小, 因此, Zipf 分布下的相对误差较小. 考虑到 η 一定时, NCBF 和 dICBF 的错误概率一定, 因此, Zipf 分布下的计数误差比均匀分布下的计数误差要小. 对于 SCBF 而言, 由于其阳性误差概率为 0.5^{k_s} , 而由公式(3)可知, Zipf 分布下的 $E(f)$ 值比均匀分布下的 $E(f)$ 值要小, 因此 Zipf 分布下的 k_s 值比均匀分布下的 k_s 要大, 故 Zipf 分布下的阳性误差概率比均匀分布下的阳性误差概率要低, 于是 Zipf 分布下的计数误差比均匀分布下的计数误差要小.

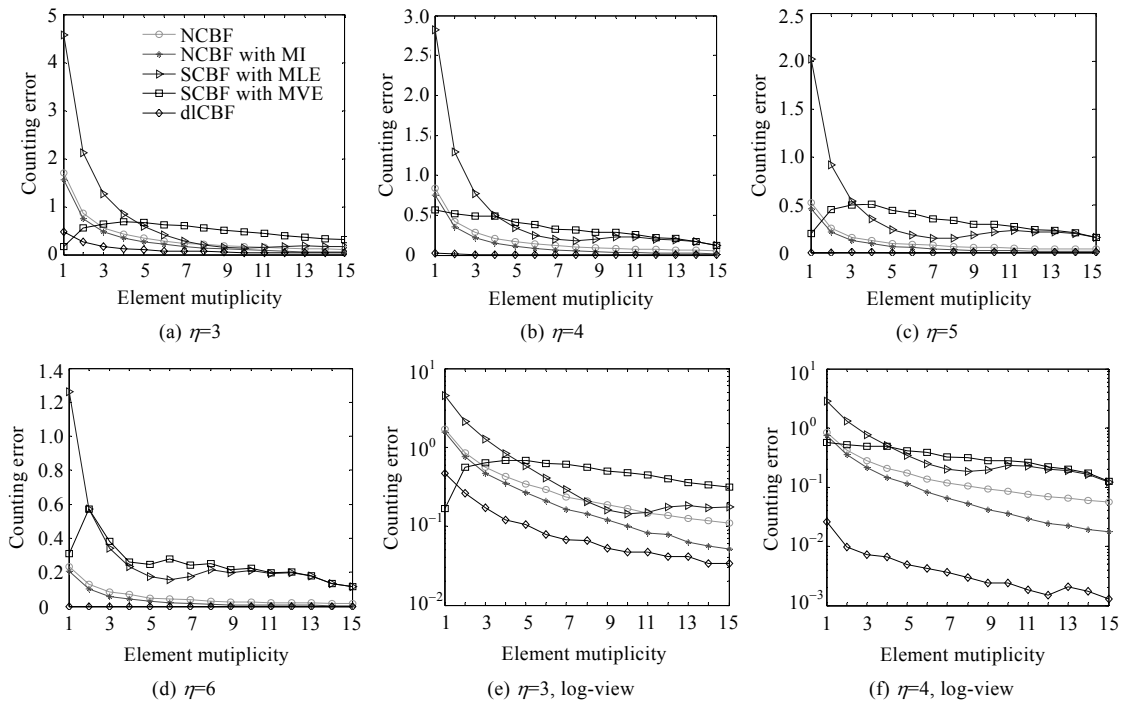


Fig.4 Counting error of three CBF schemes under various space complexity, with uniform element multiplicity distribution

图 4 当元素频率服从均匀分布时,不同空间复杂度下,3 种 CBF 的计数误差

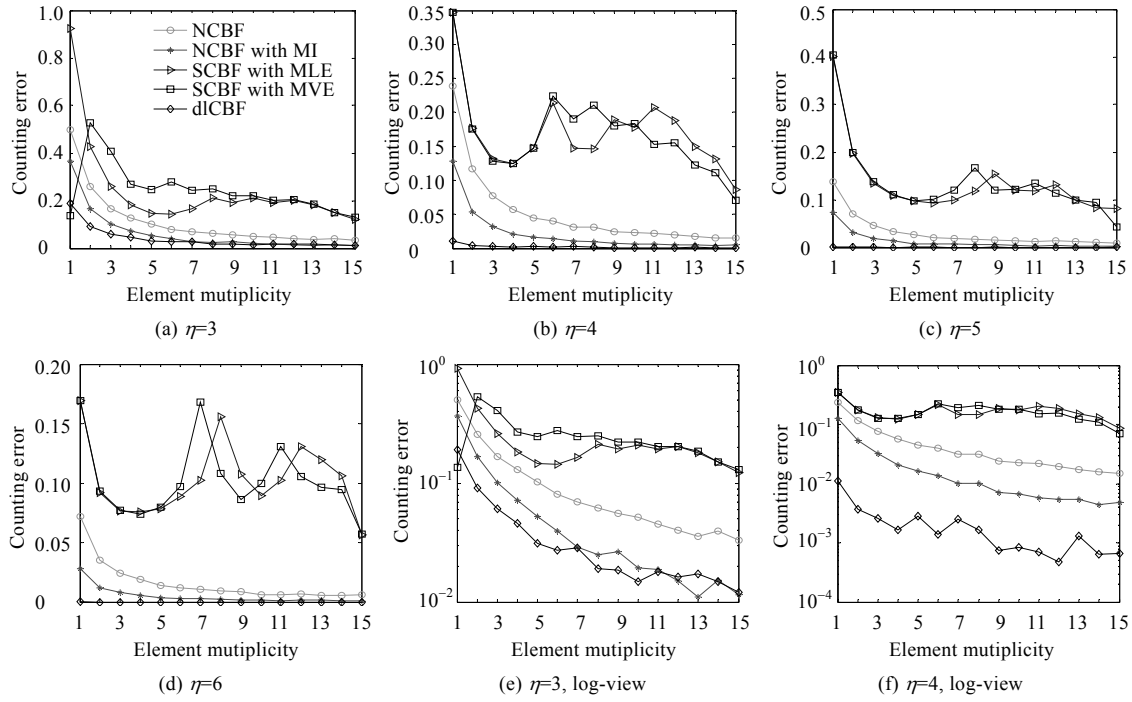


Fig.5 Counting error of three CBF schemes under various space complexity, with Zipfian element multiplicity distribution

图 5 当元素频率服从 Zipf 分布时,不同空间复杂度下,3 种 CBF 的计数误差

dlCBF 的相对较小的计数误差主要源于其相对较低的错误概率.例如,当 $\eta=4$ 时,由公式(7)和公式(8)可知, $l_d^f = 14$ 且 $b_d^m = 12$,于是由公式(1)和公式(5)可知,此时 NCBF 和 dlCBF 的错误概率分别为 0.125 和 0.002 9,两者相差两个数量级.但同时实验结果表明,此时 NCBF 和 dlCBF 的相对误差在同一个数量级,如图 6 所示.因此,我们可以得出结论,dlCBF 的相对较低的错误概率是导致其相对较小的计数误差的主要原因.

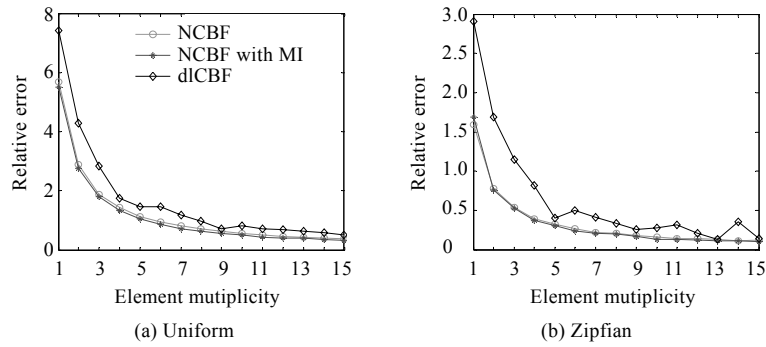


Fig.6 Relative error of NCBF and dlCBF when $\eta=4$

图 6 当 $\eta=4$ 时,NCBF 和 dlCBF 的相对误差

5.2 空间复杂度比较

表 6 给出了当元素频率服从均匀分布时,在计数误差相近的前提下,3 种 CBF 的空间复杂度的比较.从表 6 可以看出,dlCBF 的空间复杂度最低,NCBF 次之,SCBF 最高.随着 η 的增加,3 种 CBF 的空间复杂度的差距也越发

明显.以 NCBF 和 dlCBF 为例来说明则不难理解产生这一趋势的原因.由公式(1)可知,NCBF 的每个元素所需空间每增加 $w_n/\ln(2)$,NCBF 的错误概率才减小一半.同时,由公式(5)可知,当 b_d^m 一定时, l_d^f 每增加 1,dlCBF 的错误概率即可减小一半.可见,dlCBF 的空间利用率相对较高.

Table 6 Per-Element space occupation of three CBF schemes under similar counting error

表 6 计数误差相近时,3 种 CBF 的每元素所需空间

	NCBF	dlCBF	SCBF with MLE	SCBF with MVE
$\eta=3$	15	13	$1.72 \times 15 (k_s=2)$	$1.72 \times 15 (k_s=2)$
$\eta=4$	20	14	$3.16 \times 20 (k_s=4)$	$3.16 \times 20 (k_s=4)$
$\eta=5$	25	15	$4.61 \times 25 (k_s=6)$	$4.61 \times 25 (k_s=6)$
$\eta=6$	30	16	$7.49 \times 30 (k_s=10)$	$7.49 \times 30 (k_s=10)$

5.3 负载适应性比较

在比较负载适应性时,我们首先利用第 5.2 节的实验结果适当设置 4 种 CBF 的参数,使得其满负载时计数误差接近.我们按照表 6 的第 2 行进行参数设置,SCBF 采用 MLE 估计器.为了便于分析 BSdlCBF 的错误概率,我们对 dlCBF 的参数进行了调整:由表 6 的第 2 行并根据公式(7)、公式(8)所计算出的参数值为 $l_d^f = 8, b_d^m = 8$; 本节中我们将这两个参数调整为 $l_d^f = 7, b_d^m = 3$,由公式(5)可知,如此调整对 dlCBF 的错误概率影响不大.设置平均桶负载以 0.5 为步长,从 2.5 减小到 1,如此设置意味着负载率分别为 $\rho=2.5/3, \rho=2/3, \rho=1.5/3, \rho=1/3$.设置元素数量 $N=10^5$.

图 7(a)~图 7(d)为元素出现频率服从均匀分布时,4 种计数型 Bloom filter 的归一化计数误差(图 7(a)、图 7(b)中,纵坐标的范围为[0.1,2];图 7(c)、图 7(d)中,纵坐标的范围为[0.0001,2]).

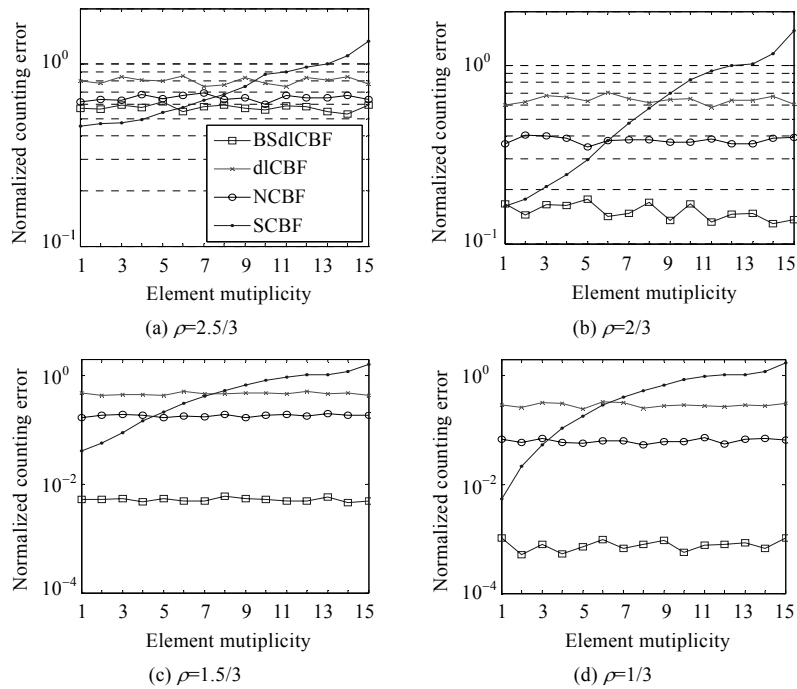


Fig.7 Normalized counting error of the four CBF schemes under various load rate, with uniform element multiplicity distribution

图 7 元素频率服从均匀分布时,不同负载率下,4 种计数型 Bloom filter 的归一化计数误差

从图 7 可以看出,随着负载率的下降,BSdlCBF 的计数误差下降速率最快,NCBF 次之,dICBF 较慢,而 SCBF 最差.另外,由图 7 还可以看出,BSdlCBF,dICBF 以及 NCBF 的不同元素频率点的归一化计数误差较为接近,但是 SCBF 不同频率点的归一化计数误差差别较大.出现上述现象的原因在于,BSdlCBF,dICBF 以及 NCBF 的错误概率与元素频率无关,而 SCBF 的错误概率与元素频率相关.在图 7 中,SCBF 小频率点的计数误差随着负载率的下降减小得较快,而大频率点的计数误差随着负载率的下降几乎没有变化.这是因为 SCBF 的计数误差同时受哈希误差和 Coupon Collector 误差^[4]的影响.前者对小频率点的计数误差影响较大,而后者对大频率点的影响较大.随着负载率的降低,哈希误差逐渐减小,因此,小频率点的计数误差下降得较为明显.但是,大频率点的计数误差由于受 Coupon Collector 误差的影响,计数误差并不会出现明显的下降.

图 8(a)~图 8(d)为当元素出现频率服从参数为 $z=1.5$ 的 Zipf 分布时,4 种计数型 Bloom filter 的归一化计数误差(图 8(a)、图 8(b)中,纵坐标的范围为 $[0.05,2]$;图 8(c)、图 8(d)中,纵坐标的范围为 $[0.0001,2]$).图 8 所呈现出的规律与图 7 一致,可见,在不同的元素频率分布下,BSdlCBF 都具有良好的负载适应性.

图 9 为负载率分别为 $2.5/3, 2/3, 1.5/3, 1/3$ 时,4 种计数型 Bloom filter 的归一化错误概率的平均值.

对于 NCBF,dICBF 和 BSdlCBF 而言,由于错误概率与元素频率无关,因而归一化错误概率的平均值实际上等于其归一化错误概率.由图 9 可以看出,随着负载率的下降,BSdlCBF 的错误概率下降速率最快,NCBF 次之,dICBF 比 NCBF 下降得慢,SCBF 下降得最慢.这与计数误差的下降规律一致.

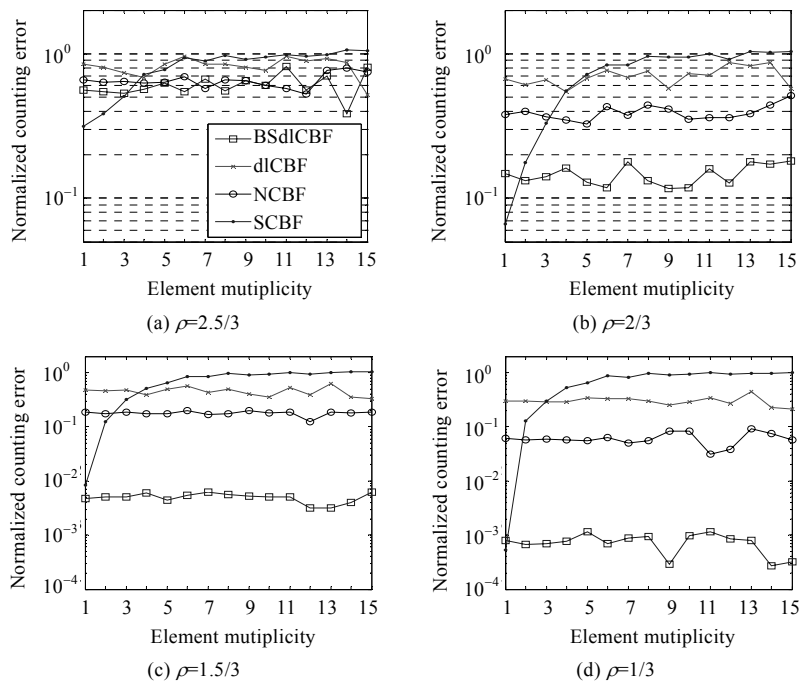


Fig.8 Normalized counting error of the four CBF schemes under various load rate, with Zipfian element multiplicity distribution

图 8 当元素频率服从 Zipf 分布时,不同负载率下,4 种计数型 Bloom filter 的归一化计数误差

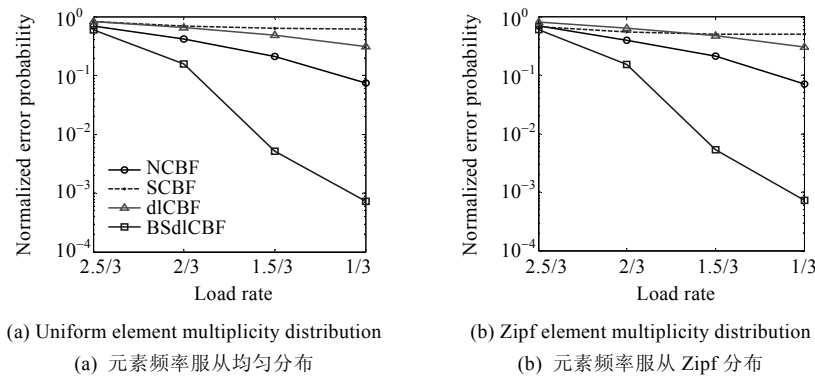


Fig.9 Mean normalized error probability of the four CBF schemes under various load rate

图 9 不同负载率下,4种计数型 Bloom filter 的平均归一化错误概率

6 相关工作

为使 Bloom filter 能够支持其所表示的集合的删除操作,Fan 等人提出了 NCBF^[2].Estan 和 Cohen 等人提出采用最小增加(minimum increase,简称 MI)的策略来改善 NCBF 的计数查询的性能^[3,7].Kumar 等人提出了 SCBF,用于骨干网流量测量^[4],但是文献[4]没有将 SCBF 与其他 CBF 作定量的性能比较.Bonomi 等人提出了 dlCBF^[6],并且比较了 dlCBF 和 NCBF 的从属查询的性能.文献[6]的实验结果表明,在相同的从属查询错误概率下,dlCBF 所需的存储空间为 NCBF 的一半或者更少.但是,文献[6]并没有对 dlCBF 和 NCBF 的计数查询的性能进行比较.

已有的比较工作中采用错误概率和空间复杂度作为衡量 CBF 性能的指标^[6].对于计数查询而言,不仅需要知道错误发生的概率,而且需要知道相对误差的大小.此外,对于硬件实现而言,负载适应性也是一项非常重要的性能指标.因此,本文以空间复杂度、计数误差以及负载适应性作为性能衡量尺度,对 NCBF,SCBF,dlCBF 以及 BSdlCBF 的性能进行了系统的比较.

7 结论

计数型 Bloom filter 是一种可支持动态集合近似计数查询的数据结构,其突出优点是计算和存储的效率高,结构简单,易于硬件实现,从而在高速数据的实时处理领域,如骨干网流量分析中有着广泛的应用.本文通过实验,以计数误差、空间复杂度和负载适应性作为性能指标,对 4 种计数型 Bloom filter 的性能进行了定量比较.实验结果表明,本文提出的 BSdlCBF 具有最小的计数误差、最低的空间复杂度和最佳的负载适应性,BSdlCBF 赢得上述性能优势的代价是其计算复杂度和空间复杂度比其他 3 种 CBF 略高.本文分析比较的 4 种 CBF 在计算复杂度、空间复杂度以及计数精度之间有不同的权衡.本文的结论为 CBF 在众多领域中的选择应用提供了最优参数设置准则以及定量的性能参考.

References:

- [1] Bloom BH. Space/Time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 1970,13(7):422-426. [doi: 10.1145/362686.362692]
- [2] Fan L, Cao P, Almeida J, Broder AZ. Summary cache: A scalable wide-area Web cache sharing protocol. *IEEE/ACM Trans. on Networking*, 2000,8(3):281-293. [doi: 10.1109/90.851975]
- [3] Cohen S, Matias Y. Spectral bloom filters. In: *Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data*. New York: ACM Press, 2003. 241-252.

- [4] Kumar A, Xu J. Space-Code bloom filter for efficient per-flow traffic measurement. In: Proc. of the IEEE INFOCOM 2004, Vol.3. Washington: IEEE Computer Society Press, 2004. 1762–1773.
- [5] Pagh A, Pagh R, Rao S. An optimal bloom filter replacement. In: Proc. of the 16th Annual ACM-SIAM Symp. on Discrete Algorithms. New York: ACM Press, 2005. 823–829.
- [6] Bonomi F, Mitzenmacher M, Panigrahy R, Singh S, Varghese G. An improved construction for counting bloom filters. In: Proc. of the 14th Conf. on Annual European Symp., Vol.14. Springer-Verlag, 2006. 684–695.
- [7] Estan C, Varghese G. New directions in traffic measurement and accounting. ACM SIGCOMM Computer Communication Review, 2002,32(4):323–336. [doi: 10.1145/964725.633056]
- [8] Broder A, Mitzenmacher M. Network applications of bloom filters: A survey. Internet Mathematics, 2004,1(4):485–509.
- [9] Bonomi F, Mitzenmacher M, Panigrahy R, Singh S, Varghese G. Beyond bloom filters: From approximate membership checks to approximate state machines. ACM SIGCOMM Computer Communication Review, 2006,36(4):315–326. [doi: 10.1145/1151659.1159950]
- [10] NLANR. Passive measurement and analysis (PMA). 2004. <http://pma.nlanr.net>
- [11] Vocking B. How asymmetry helps load balancing. Journal of the ACM, 2003,50(4):568–589. [doi: 10.1145/792538.792546]
- [12] Broder A, Mitzenmacher M. Using multiple hash functions to improve IP lookups. In: Proc. of the IEEE INFOCOM, Vol.3. Washington: IEEE Computer Society Press, 2001. 1454–1463.
- [13] Cheng G, Gong J, Ding W, Xu JL. A hash algorithm for IP flow measurement. Journal of Software, 2005,16(5):652–658 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/652.htm> [doi: 10.1360/jos160652]
- [14] Mitzenmacher M, Vocking B. The asymptotics of selecting the shortest of two, improved. In: Suhov Y, ed. Proc. of the Analytic Methods in Applied Probability: In Memory of Fridrikh Karpelevich. New York: American Mathematical Society, 2003.
- [15] Mitzenmacher M. Studying balanced allocations with differential equations. In: Proc. of the Combinations, Probability, and Computing. Cambridge: Cambridge University Press, 1999. 473–482.
- [16] Xu SL. A Collection of C Utility Routines. 3rd ed., Beijing: Tsinghua University Press, 2004 (in Chinese).

附中文参考文献:

- [13] 程光, 龚俭, 丁伟, 徐加羚. 面向 IP 流测量的哈希算法研究. 软件学报, 2005, 16(5): 652–658. <http://www.jos.org.cn/1000-9825/16/652.htm> [doi: 10.1360/jos160652]
- [16] 徐士良. 常用算法程序集(C语言描述). 第3版, 北京: 清华大学出版社, 2004.

附录 A

证明: 为方便起见, 令 $n=N, k=k_n, m=m_n$. 对于任一计数器 C, C 中累积包含了 i 个元素的频率值的概率 $P(i)$ 服从参数为 $\left(n \cdot k, \frac{1}{m}\right)$ 的二项分布, 即

$$P(i) = C_{nk}^i \left(\frac{1}{m}\right)^i \left(1 - \frac{1}{m}\right)^{nk-i} \quad (19)$$

考虑到 $k = \left\lceil \frac{m}{n} \cdot \ln 2 \right\rceil$, 即 $n \cdot k = \lceil \ln 2 \cdot m \rceil$, 于是有:

$$P(i) = C_{\lceil \ln(2) \cdot m \rceil}^i \left(\frac{1}{m}\right)^i \left(1 - \frac{1}{m}\right)^{\lceil \ln(2) \cdot m - i \rceil} \quad (20)$$

对于任意计数器 C, C 中包含至少 3 个元素的累积频率值的概率为 $P(c \geq 3F) = 1 - P(0) - P(1) - P(2)$. 考虑到 m 较大, 且 $\lim_{m \rightarrow \infty} (P(0)) = 1/2, \lim_{m \rightarrow \infty} (P(1)) = \ln(2)/2, \lim_{m \rightarrow \infty} (P(2)) = (\ln(2)/2)^2$, 于是有:

$$P(c \geq 3F) \approx 1 - 1/2 - \ln(2)/2 - (\ln(2)/2)^2 \quad (21)$$

当设置 $w_n = 2F$ 时, NCBF 产生阴性错误的概率为 $P_{fn} = (P(c \geq 3F))^k$. 而考虑到 NCBF 产生阳性错误的概率 $P_{fp} = (1/2)^k$, 于是有:

$$P_{fn}/P_{fp}=(2 \cdot P(c \geq 3F))^k < 0.07^k \tag{22}$$

证毕. □

附录 B

本附录说明如何在已知 *d*-left 哈希函数的平均桶负载 *r* 的条件下确定其桶负载分布,即确定对于任意一个桶,其中有 *i* 个元素的概率,即 $P(r,i)$,其中 $0 \leq i \leq b$.

令 $X_j(r,i)$ 为平均桶负载为 *r* 时,第 *j* 个块中至少有 *i* 个元素的桶在所有桶中所占的比例,其中 $1 \leq j \leq d, 0 \leq i \leq b$. 为便于表述,令 $k=i \cdot d+j$ 且 $Y_k(r)=X_j(r,i)$. Mitzenmacher 等人指出,当桶的数量和桶深都趋于无穷大时, $Y_k(r)$ 可以用下列微分方程组来描述^[14,15]:

$$\frac{dy_k}{dr} = d^d (y_{k-d} - y_k) \prod_{s=k-d+1}^{k-1} y_s \tag{23}$$

$\forall k, y_k(0)=0$. 上述微分方程组可以通过常用的数值方法进行求解. 一般地,当桶的数量和桶深都有限时,方程组 (23) 也能极佳地逼近 $Y_k(r)$ ^[12,15].

令 $P'(r,i)$ 为平均桶负载为 *r* 的条件下,对于 *d*-left 哈希函数的任意一个桶,其中至少有 *i* 个元素的概率. 于是

$$P'(r,i) = \sum_{j=1}^d y_{id+j}(r) \tag{24}$$

显然, $P'(0,i)=1$. 于是有:

$$P(r,i) = P'(r,i) - P'(r,i+1) \tag{25}$$

这样即可求得 $P(r,i)$.

表 7 给出了当 $d=4, b=8$ 时的一个数值例子. 平均桶负载 *r* 以 0.5 为步长,从 0 增加到 6. 微分方程组 (23) 采用定步长欧拉方法^[16]进行求解.

Table 7 Distribution of number of elements in a bucket under various bucket load when $d=4$

表 7 当 $d=4$ 时,不同平均桶负载下,桶中元素数量的分布

		<i>l</i>								
		0	1	2	3	4	5	6	7	8
<i>r</i>	0.5	5.010e-1	4.980e-1	9.850e-4	3.317e-41	0	0	0	0	0
	1	1.242e-1	7.515e-1	1.242e-1	7.763e-11	6.822e-143	0	0	0	0
	1.5	1.896e-2	4.637e-1	5.157e-1	1.623e-3	6.488e-38	0	0	0	0
	2	2.612e-3	1.227e-1	7.468e-1	1.279e-1	1.214e-10	4.816e-140	0	0	0
	2.5	3.545e-4	1.959e-2	4.614e-1	5.170e-1	1.676e-3	1.051e-37	0	0	0
	3	4.802e-5	2.800e-3	1.225e-1	7.464e-1	1.283e-1	1.263e-10	8.652e-140	0	0
	3.5	6.503e-6	3.931e-4	1.964e-2	4.612e-1	5.171e-1	1.682e-3	1.107e-37	0	0
	4	8.806e-7	5.501e-5	2.820e-3	1.225e-1	7.463e-1	1.283e-1	1.269e-10	9.276e-140	0
	4.5	1.192e-7	7.688e-6	3.980e-4	1.965e-2	4.612e-1	5.171e-1	1.682e-3	1.114e-37	0
	5	1.615e-8	1.073e-6	5.602e-5	2.822e-3	1.225e-1	7.463e-1	1.283e-1	1.270e-10	9.363e-140
	5.5	2.186e-9	1.497e-7	7.881e-6	3.986e-4	1.965e-2	4.612e-1	5.171e-1	1.682e-3	1.115e-37
	6	2.960e-10	2.086e-8	1.108e-6	5.617e-5	2.822e-3	1.225e-1	7.463e-1	1.283e-1	1.270e-10

附录 C

证明:为方便起见,令 $b = b_d^m, l = l_d^f, R = l_d^m, C = l_d$, 则原问题转化为:若 $\frac{b+1}{b} \cdot (l + R) = C$, 其中 *R* 和 *C* 为常数,求 *l* (或者 *b*) 的最优取值,使得 $\frac{b}{2^l}$ 最小.

由假设可知:

$$b = \frac{R+l}{C-R-l} \quad (26)$$

于是,我们需要寻找 l 的最优取值使得 f 最小,其中,

$$f = \frac{C-R-l}{R+l} \cdot \frac{1}{2^l} \quad (27)$$

对 f 求导并令导数为 0,有:

$$\frac{df}{dl} = \frac{1}{2^l} \left(\frac{C}{(C-R-l)^2} - \ln(2) \cdot \frac{R+l}{C-R-l} \right) = 0 \quad (28)$$

于是有:

$$(l+R) \cdot (C-R-l) \cdot \ln(2) = C \quad (29)$$

求解公式(29)的方程,可得:

$$l = \frac{1}{2} \cdot \left(C - 2R \pm \sqrt{C^2 - \frac{4C}{\ln(2)}} \right) \quad (30)$$

将公式(30)代入公式(26)可得:

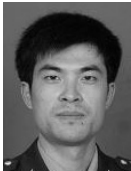
$$b = \left(C \pm \sqrt{C^2 - \frac{4C}{\ln(2)}} \right) / \left(C \mp \sqrt{C^2 - \frac{4C}{\ln(2)}} \right) \quad (31)$$

考虑到 $b \geq 1$,因此公式(31)分子上取加号,分母上取减号.同时考虑到 l 和 b 为整数,于是有:

$$l = \left\lceil \frac{1}{2} \cdot \left(C - 2R + \sqrt{C^2 - \frac{4C}{\ln(2)}} \right) \right\rceil \quad (32)$$

$$b = \left\lceil \left(C + \sqrt{C^2 - \frac{4C}{\ln(2)}} \right) / \left(C - \sqrt{C^2 - \frac{4C}{\ln(2)}} \right) \right\rceil \quad (33)$$

将公式(32)和公式(33)中的符号换回,即可得定理 1. □



张进(1979—),男,江苏镇江人,博士,工程师,主要研究领域为网络测量,无线网络,片上系统(SoC).



刘勤让(1975—),男,博士,副教授,主要研究领域为业务识别与控制,网络服务质量.



邬江兴(1953—),男,教授,博士生导师,中国工程院院士,主要研究领域为信息网络与交换技术.