

计数布鲁姆过滤器代数运算

田小梅^{1),2)} 张大方¹⁾ 谢 鲲¹⁾ 史长琼^{1),3)} 杨晓波¹⁾

¹⁾(湖南大学信息科学与工程学院 长沙 410082)

²⁾(湖南环境生物职业技术学院信息技术系 湖南 衡阳 421005)

³⁾(长沙理工大学计算机与通信工程学院 长沙 410114)

摘 要 文中探讨计数布鲁姆过滤器的代数运算和集合运算的一致性关系,研究使用计数布鲁姆过滤器代数运算进行集合成员查询的性能.理论分析和实验结果表明,计数布鲁姆过滤器的并、交、补、减、异或运算产生的新过滤器依然保持计数布鲁姆过滤器的特征,支持元素的删除操作,不会出现假阴性,能用于集合并集、交集、补集、差集及对称差的成员查询;当使用两个原始的计数布鲁姆过滤器查询补集、差集及对称差元素时,会存在部分本来属于补集、差集或对称差的元素被判为不属于补集、差集或对称差的问题,而使用计数布鲁姆过滤器代数运算后的过滤器进行补集、差集及对称差成员查询,则不存在上述问题,空间效率能提高一倍,时间效率亦能显著地得到改善.计数布鲁姆过滤器代数运算的使用有利于进一步扩展计数布鲁姆过滤器的应用范围.譬如计数布鲁姆过滤器减运算可用作一种新的集合调和的方法,用于分布式系统中大型文件的分发.

关键词 代数运算;计数布鲁姆过滤器;集合调和;计算机网络;分布式计算

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2012.02598

Algebra Operations on Counting Bloom Filters

TIAN Xiao-Mei^{1),2)} ZHANG Da-Fang¹⁾ XIE Kun¹⁾ SHI Chang-Qiong^{1),3)} YANG Xiao-Bo¹⁾

¹⁾(School of Information Science and Engineering, Hunan University, Changsha 410082)

²⁾(Department of Information Technology, Hunan Environment-Biological Polytechnic, Hengyang, Hunan 421005)

³⁾(School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114)

Abstract This paper examines the consistence between algebra operations on counting Bloom filters and algebra operations on data sets, and studies the membership query performances of algebra operations on counting Bloom filters. Theoretical analyses and simulations show that the counting Bloom filter which is ORed (ANDed, COMPLEMENTed, SUBTRACTed, XORed) from the original counting Bloom filters can support membership query on data set ORed (ANDed, COMPLEMENTed, SUBTRACTed, XORed) from the original data sets. When using the two original counting Bloom filter to query elements belonged to complementary set, differences or symmetric differences of the two sets, some complementary set elements, differences or symmetric differences of the sets will be misjudged, while the query method using algebra operations on counting Bloom filters has no false negatives and gain a remarkable improvement in space complexity and time complexity over the method using double independent counting Bloom filters, hence it can be applied to various network fields. For example, SUB operation can be used in complete set reconciliation, which is a new set reconciliation, for distributed database and file systems.

Keywords algebra operations; counting Bloom filter; set reconciliation; computer network; distributed computing

收稿日期:2010-05-12;最终修改稿收到日期:2011-10-13. 本课题得到国家自然科学基金(61003305,60703097,90718008)、国家“九七三”重点基础研究发展规划项目基金(2007CB310702)、湖南省教育厅资助科研项目(08D055)资助. 田小梅,女,1971年生,博士研究生,副教授,主要研究方向为P2P网络、分布式计算. E-mail: xiaomeit@163.com. 张大方,男,1959年生,博士,教授,博士生导师,主要研究领域为可信系统与网络、网络测试、软件测试、软件容错. 谢 鲲,女,1978年生,博士,副教授,主要研究方向为可信系统与网络、无线网络、算法设计. 史长琼,女,1968年生,博士研究生,副教授,主要研究方向为网络信息安全、人工智能. 杨晓波,女,1973年生,博士研究生,讲师,主要研究方向为可信系统与网络、人工智能、软件设计.

1 引言

自 Burton Bloom 在 20 世纪 70 年代提出之后,布鲁姆过滤器已被广泛用于拼写检查和数据库系统中.近一二十年,伴随着计算机网络的普及和发展,布鲁姆过滤器在网络领域获得了新生,各种布鲁姆过滤器算法的变种和新的应用不断出现^[1-11].

按布鲁姆过滤器向量空间的构成来看,有两类基本的布鲁姆过滤器:位向量布鲁姆过滤器和计数器向量布鲁姆过滤器.位向量布鲁姆过滤器采用位向量表示数据集合,能支持集合元素的插入、查询操作,如标准布鲁姆过滤器(Bloom Filter, BF)^[3]、动态布鲁姆过滤器(Dynamic Bloom Filter, DBF)^[5]等.计数器向量布鲁姆过滤器使用计数器向量表示数据集合,既能支持集合元素的插入、查询操作,也能支持元素的删除操作,如计数布鲁姆过滤器(Counting Bloom Filter, CBF)^[4]、光谱布鲁姆过滤器(Spectral Bloom Filter, SBF)^[6]等.在使用布鲁姆过滤器查询一个元素是否属于某个集合时,会有一定的假阳性(False Positive),也就是说,有可能把不属于这个集合的元素判为属于这个集合,但不存在假阴性(False Negative),即不会把属于这个集合的元素判为不属于这个集合^[1-2].

有时,我们不仅需要查询元素是否属于某集合,还需要查询元素是否不属于某特定集合(即查询某元素是否属于该特定集合的补集,简称补集查询),或者查询元素是否属于集合的交集、并集、差集及对称差.

譬如,分布式 P2P 系统的多关键字搜索(Multi-keyword Search)模型中最常见的两类查询:And 查询、Or 查询,这两类查询分别对应求集合交集问题与求集合并集问题;当使用分布式倒排索引技术并用 BF 表示与关键字匹配的文档列表来实现多关键字搜索时,两类查询则分别对应 BF 的交运算与并运算,如文献[12],在处理跨节点连接查询时传递 BF 表示的文档匹配列表,并进行 BF 的并运算或交运算,以减少信息传输量,提高检索效率.事实上,在多关键字搜索时,还有一类减查询常在各类搜索引擎中出现,如查询包含关键字 X 但不包含关键字 Y 的网页,此时可以使用布鲁姆过滤器的减运算来支持减查询.

又如,在分布式内容分发系统中,假如要将一

个大文件从源节点分发到诸多对等节点,节点(设为 A)在从源节点下载某些文件块的同时可从其它对等节点(设为 B)下载其缺失的文件块,以有效地提高下载速率.这种情况下,首先要找出 B 中有而 A 尚未接收到的文件块(即集合的差),然后在节点间仅传送这部分差集文件块即可,免去传送所有文件块的网络开销.此时,通常文件块用布鲁姆过滤器表示,需要借助布鲁姆过滤器来找出集合的差集元素,即差集文件块.

集合用布鲁姆过滤器(或计数布鲁姆过滤器)表示后,我们首先想到的方法是使用两个单独的过滤器来查询元素是否属于并集、交集、补集、差集或对称差,称双过滤器查询法.如,双过滤器补集查询法,直接使用集合 S 的过滤器和全集 U 的过滤器来查询元素是否属于 $\bar{S}$,其查询逻辑如下:首先使用全集 U 的过滤器来查询元素是否在 U 中,如果查询结果为否,则判定元素不属于 $\bar{S}$;如果查询结果为是,则继续使用集合 S 的过滤器查询元素是否属于 S,如果查询结果仍然是肯定的,则认为该元素不属于集合的补集 $\bar{S}$,否则,认定该元素属于集合的补集 $\bar{S}$.在该过程中,由于查询存在如下可能:少量不属于 S 的元素被判为属于 S,即少量原本属于 $\bar{S}$ 的元素被判为不属于 $\bar{S}$,或者说,在这一直接使用过滤器进行补集查询的过程中出现了假阴性.事实上,两集合用布鲁姆过滤器表示后,如果直接使用两个原始的布鲁姆过滤器查询元素是否属于差集、对称差等,存在与补集查询同样的问题,除存在假阳性外,还存在假阴性,极大地影响查询结果的可靠性,而且,使用两个独立的布鲁姆过滤器查询差集、对称差元素,需对两个过滤器向量进行查找操作,空间效率及时间效率均低.如果使用双过滤器查询法进行并集或交集查询,虽然不存在假阴性,但是需要保存两个单独的过滤器向量,而且查询时需要在两个过滤器向量上进行定位、比较操作,从而影响查询算法的时间、空间效率.

由于使用双过滤器法查询补集、差集或对称差元素时存在着前述假阴性问题,因此,我们尝试从布鲁姆过滤器向量运算的角度寻求能解决前述假阴性问题的方法.

目前研究布鲁姆过滤器向量运算及应用的工作为数不多^[5,13-14].文献[13]指出可用布鲁姆过滤器位向量按位或、按位与得到的向量分别表示并集、交集.文献[5]定义了标准布鲁姆过滤器(BF)

的并运算和交运算以及动态布鲁姆过滤器、多维布鲁姆过滤器、多维动态布鲁姆过滤器等位向量布鲁姆过滤器的并运算,且运用 BF 的并运算构造了一个基于标准布鲁姆过滤器的知情搜索协议(Informed search protocol based on Bloom filters).文献[14]研究位向量布鲁姆过滤器代数运算和集合查询的关系,得出结论如下:位向量布鲁姆过滤器的并、交运算能够支持集合并集、交集的元素查询,而 BF 补、减、异或运算则由于存在较大的假阴性,不能用于集合补集、差集、对称差的元素查询;直接通过并、交运算后的位向量布鲁姆过滤器完成查询操作,与直接使用两个原始的布鲁姆过滤器相比较,可节约位向量的存储空间及位向量的查询次数,提高查询效率。

现有工作研究的是位向量布鲁姆过滤器的代数运算,研究结论表明:位向量布鲁姆过滤器并、交运算的使用可推广布鲁姆过滤器的应用范围,简化使用布鲁姆过滤器的系统设计^[5,14-15].但是, BF 的减、异或、补运算不能用于差集、对称差、补集的元素查询^[14],而 CBF 支持元素的删除操作,计数器值可实现减法运算, CBF 的应用优势除可用于动态更新的应用场合^[4]外,是否还具有其它方面的优势呢?譬如,是否可对 CBF 计数器向量进行并、交、减、异或及补运算? CBF 的减、异或及补运算是否能用于差集、对称差及补集的元素查询?这就是本文的研究目的。

本文通过合理地定义计数器向量的并、交、补、减、异或代数运算,使得代数运算后的 CBF 均能较准确地反映相应集合,不会出现假阴性,可以使用 CBF 的并(交、补、减、异或)运算进行集合的并集(交集、补集、差集、对称差)查询;使用 CBF 代数运算进行补集(差集、对称差)查询,与使用两个原始的 CBF 进行查询的方法相比,不存在假阴性,空间效率能提高一倍,时间效率亦能显著地得到改善.本文以 CBF 减运算为例,将之应用于分布式计算领域中的集合调和过程,由于 CBF 减运算后得到的过滤器在查询差集时不存在假阴性,使用该过滤器查询时能找出所有差集元素,从而得到完整的集合并集.仿真实验证明,使用 CBF 减运算的集合调和法是一种简单、可靠、高效的调和方法. CBF 减运算在分布式集合调和中的成功应用,说明本文的 CBF 代数运算在分布式系统中具有广泛的应用潜力。

本文第 2 节概述计数布鲁姆过滤器算法;第 3

节定义计数布鲁姆过滤器并、交、补、减、异或运算;第 4 节对计数布鲁姆过滤器代数运算和集合运算的关系作了数学分析和证明;第 5 节是计数布鲁姆过滤器代数运算的性能分析及模拟实验部分;第 6 节是计数布鲁姆过滤器代数运算的应用实例——使用 CBF 减运算的集合调和;第 7 节是本文工作的总结及下一步工作重点的介绍。

2 计数布鲁姆过滤器查询算法

2.1 标准布鲁姆过滤器

标准布鲁姆过滤器^[3]查询算法核心是一个长度为 m 的位向量和一个由 k 个 Hash 函数构成的 Hash 函数组. n 个元素的数据集合 $S = \{s_1, s_2, \dots, s_n\}$ 通过 k 个 Hash 函数 $h_1, h_2, \dots, h_k$ 映射到位向量 $\mathbf{BF} = (b_1 b_2 b_3 \dots b_m)$ 中. 位向量 $\mathbf{BF}$ 就是集合 S 的布鲁姆过滤器表示,记为 $\mathbf{BF}(S)^{m,k}$,或简记为 $\mathbf{BF}(S)$. 各 Hash 函数相互独立且函数的取值范围为 $\{0, 1, 2, \dots, m-1\}$ 在初始化时, $\mathbf{BF}$ 的每位都为 0. 当元素 x 插入集合时,计算 x 对应的 k 个位串索引值 $h_1(x), h_2(x), \dots, h_k(x)$ (即 Hash 地址)将 $\mathbf{BF}$ 向量相应位置 1 ($b_{h_j(x)} = 1, 1 \leq j \leq k$),完成元素的插入. 查询给定的元素 x 是否属于集合 S 首先计算 x 对应的 k 个位串索引值 ($h_1(x), h_2(x), \dots, h_k(x)$),然后检查向量 $\mathbf{BF}$ 的 k 个位置是否全为 1,若全 1,则认为 x 属于集合 S 中;否则,只要此 k 个位置中任一为 0,则 x 必定不属于集合 S . 标准布鲁姆过滤器存在假阳性,可能将不属于集合的元素判成属于集合,其假阳性概率为^[1,3]

$$f_{p_{m,n,k}}^{\mathbf{BF}} = (1 - e^{-kn/m})^k.$$

当集合规模 n 和向量空间 m 一定的情况下,使布鲁姆过滤器查询假阳性概率达最小的 k 值为

$$k = (\ln 2)(m/n).$$

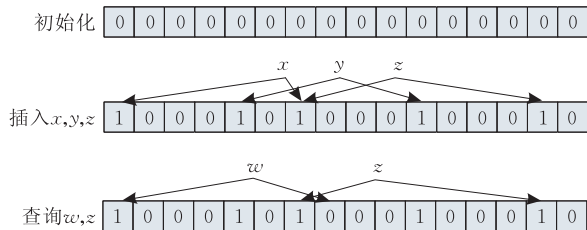


图 1 标准布鲁姆过滤器算法

使用标准布鲁姆过滤器完成集合存储 m 比特的位向量可以表示 n 个元素的集合,每个元素平均使用 m/n 位,大大节约存储空间.使用标准布鲁姆

过滤器完成单个元素的成员查询,需进行 k 次 Hash 函数的计算,并进行 k 次位的查询,其时间复杂度为 $O(k)$,与集合中元素个数 n 无关. 布鲁姆过滤器空间简洁、查询效率高的特点,正是其得以广泛应用的原因.

标准布鲁姆过滤器的一个缺点是不支持集合元素的删除操作. 由于标准布鲁姆过滤器不处理碰撞,有可能多个元素都映射到了同一 Hash 位,因此如果执行删除操作将此 Hash 位清零,会出现假阴性,将原本属于集合的元素判定为不属于集合. 从集合的布鲁姆过滤器表示算法不难看出,集合同它的布鲁姆过滤器表示所蕴涵的信息最初是一致的,此时不会发生假阴性判定. 当布鲁姆过滤器执行了对数据元素的不正确删除操作时,破坏了集合同其布鲁姆过滤器表示之间的一致性,从而出现假阴性判定.

2.2 计数布鲁姆过滤器

计数布鲁姆过滤器^[4]将标准布鲁姆过滤器位向量的每一位扩展为一个 r 位的计数器,在插入元素时将对应的 k 个计数器的值分别加 1,删除元素时将对应的 k 个计数器的值分别减 1. n 个元素的数据集合 $S = \{s_1, s_2, \dots, s_n\}$ 通过 k 个 Hash 函数 $h_1, h_2, \dots, h_k$ 映射到由 m 个计数器组成的计数器向量 $\mathbf{CBF} = (C_1 C_2 C_3 \dots C_m)$ 中. 计数器向量 $\mathbf{CBF}$ 就是集合 S 的计数布鲁姆过滤器表示,记为 $\mathbf{CBF}(S)^{m,k}$,或简记为 $\mathbf{CBF}(S)$. 在初始化时, $\mathbf{CBF}$ 的每个计数器都为 0. 当元素 x 插入集合时,计算 x 对应的 k 个计数器索引值 $h_1(x), h_2(x), \dots, h_k(x)$ (即 Hash 地址)将 $\mathbf{CBF}$ 向量相应计数器加 1 ($C_{h_j(x)}++$, $1 \leq j \leq k$),完成元素的插入. 查询给定的元素 x 是否属于集合 S ,首先计算 x 对应的 k 个计数器索引值 $h_1(x), h_2(x), \dots, h_k(x)$,然后检查计数器向量 $\mathbf{CBF}$ 对应 k 个计数器 ($C_{h_j(x)}$, $1 \leq j \leq k$) 的值,若全大于等于 1,则认为 x 属于集合 S 中;否则,只要此 k 个计数器中任一计数器为 0,则 x 必定不属于集合 S . 当要从计数布鲁姆过滤器删除元素 x 时,计算出 x 对应的 k 个计数器索引值后将 $\mathbf{CBF}$ 向量中相应计数器进行减 1 运算 ($C_{h_j(x)}--$, $1 \leq j \leq k$) 即可.

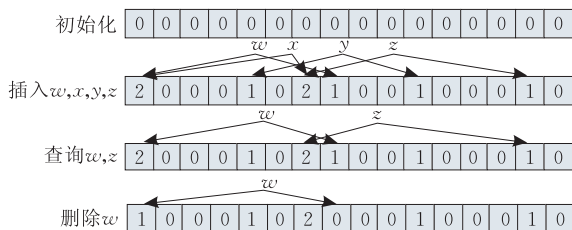


图 2 计数布鲁姆过滤器算法

由于计数布鲁姆过滤器执行删除操作时不是将元素映射的计数器采取清零操作,而是将相应计数器减 1,不会影响映射到同一 Hash 地址的其它元素的判定,从而不会出现假阴性. 计数布鲁姆过滤器通过占用 r 倍于标准布鲁姆过滤器的存储空间代价,使得布鲁姆过滤器除了支持元素的插入、查询操作外,还能支持元素的删除操作.

因为在元素插入时,会使计数布鲁姆过滤器对应计数器加 1,而计数器是由有限的比特位组成,因此可能会造成计数器溢出. 为避免计数器溢出,应选择适当的计数器位数 r . 下面通过计算溢出概率来获得通用的计数器长度设计.

我们先计算第 i 个计数器增加 j 次的概率:

$$P(C_i = j) = \binom{nk}{j} \left(\frac{1}{m}\right)^j \left(1 - \frac{1}{m}\right)^{nk-j}.$$

上式右端的表达式中,第 1 个分量表示从 nk 次 Hash 中选择 j 次,第 2 个分量表示 j 次 Hash 都选中了第 i 个计数器,第 3 个分量表示其它 $nk-j$ 次 Hash 都没有选中第 i 个计数器. 因此,第 i 个计数器的值大于或等于 j 的概率可以限定为

$$P(C_i \geq j) \leq \binom{nk}{j} \left(\frac{1}{m}\right)^j \leq \left(\frac{enk}{jm}\right)^j.$$

当 m/n 值固定时, k 的最优值为 $(\ln 2)m/n$,如果限制 $k \leq (\ln 2)m/n$,就可以得到任意计数器值大于或等于 j 的概率:

$$P(\text{Max}(C_i) \geq j) \leq m \left(\frac{e \ln 2}{j}\right)^j$$

或

$$P(\text{Max}(C_i) \geq 2^r) \leq m \left(\frac{e \ln 2}{2^r}\right)^{2^r}.$$

如果 $r=4$,那么当计数器的值达到 16 时就会溢出. 计数器溢出概率为

$$P(\text{Max}(C_i) \geq 16) \leq 1.37 \times 10^{-15} \times m.$$

由上式可知,溢出概率非常小,对于大多数应用程序来说,取 $r=4$ 就足够了^[4]. 因此,本文的计数布鲁姆过滤器使用 4 位计数器.

计数布鲁姆过滤器同样存在假阳性,其假阳性概率与标准布鲁姆过滤器的相同,为

$$f_{P_{m,n,k}}^{\mathbf{CBF}} = (1 - e^{-kn/m})^k.$$

计数布鲁姆过滤器查询算法的出现解决了标准布鲁姆过滤器不支持元素的删除这一问题,同时依然保留标准布鲁姆过滤器实现简单、查询效率高的优点.

由于计数布鲁姆过滤器具有支持元素删除操作

的优点,自其提出后,涌现了很多计数布鲁姆过滤器的扩展算法和具体应用实例.譬如,光谱布鲁姆过滤器^[6]扩展了计数布鲁姆过滤器,除支持成员查询外还支持频数查询,解决了计数器大小变动带来的动态存储问题,但是引入了复杂的索引结构,每个计数器的访问变得复杂而耗时.为此,文献[7]提出了动态计数过滤器,既支持元素出现频数查询,结构又相对简单.文献[11]则旨在进一步减少计数布鲁姆过滤器所占存储空间,研究计数器溢出条件,运用层次化设计思想和哈夫曼编码,设计出新的多层压缩计数布鲁姆过滤器以适用于网络处理器的应用场合.文献[16]中的 d-left 计数布鲁姆过滤器使用 d-left hashing 解决 Hash 表的负载平衡问题.计数布鲁姆过滤器即使分配较大的 Hash 空间,由于按最坏情况分配计数器容量,造成大量 Hash 空间闲置.使用 d-left hashing 后可以让存储的信息分布均匀,更加紧凑,从而用更小的空间(存储空间可减少 2 倍以上)存储同样多的信息.

3 计数布鲁姆过滤器代数运算

定义 1(元素的计数布鲁姆过滤器表示). 对于全集 U 中的任意元素 x ,通过 k 个 Hash 函数,映射到 m 个计数器(位数为 r)的计数布鲁姆过滤器向量 $CBF^{m,k}(CBF_1^{m,k}, CBF_2^{m,k}, \dots, CBF_i^{m,k}, \dots, CBF_m^{m,k})$ 后,生成新向量 $(CBF(x)_1^{m,k}, \dots, CBF(x)_i^{m,k}, \dots, CBF(x)_m^{m,k})$,记为 $CBF(x)^{m,k}$,其中第 i 个计数器的值变化如下:

$$CBF(x)_i^{m,k} = \begin{cases} (CBF_i^{m,k} + 1), & i = h_j(x), j = 1, \dots, k, i = 1, \dots, m \\ CBF_i^{m,k}, & \text{其它} \end{cases}$$

定义 2(集合的计数布鲁姆过滤器表示). 对于全集 U 中的任意子集 $S = \{s_1, s_2, \dots, s_n\}$, S 中所有元素都通过 k 个 Hash 函数映射到计数器个数为 m 的计数布鲁姆过滤器向量中,向量记为 $CBF(S)^{m,k}$,其中第 i 个计数器

$$CBF(S)_i^{m,k} = CBF(s_1)_i^{m,k} + \dots + CBF(s_n)_i^{m,k},$$

$i = 1, \dots, m.$

定义 3(同源计数布鲁姆过滤器). 对于全集 U 中的任意两子集 S_1 和 S_2 ,其计数布鲁姆过滤器表示分别为 $CBF(S_1)^{m,k}, CBF(S_2)^{m,k}$,若它们使用的 k 个 Hash 函数为同一组 Hash 函数,计数器个数 m 相同,且计数器大小 r 相等,则称两计数布鲁姆过滤器 $CBF(S_1)^{m,k}, CBF(S_2)^{m,k}$ 同源.下文讨论的均

为同源计数布鲁姆过滤器.

定义 4(同源计数布鲁姆过滤器的补运算). 对于计数布鲁姆过滤器 $CBF(S)^{m,k}$,其补运算 $\overline{CBF(S)^{m,k}} = \{C_i | C_i = CBF(U)_i^{m,k} - CBF(S)_i^{m,k}, i = 1, \dots, m\}$, U 为全集.

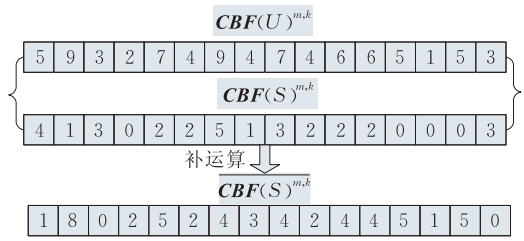


图 3 计数布鲁姆过滤器补运算

定义 5(同源计数布鲁姆过滤器的交运算). 对于两同源计数布鲁姆过滤器 $CBF(S_1)^{m,k}$ 与 $CBF(S_2)^{m,k}$,其交运算 $CBF(S_1)^{m,k} \cap CBF(S_2)^{m,k} = \{C_i | C_i = \min(CBF(S_1)_i^{m,k}, CBF(S_2)_i^{m,k}), i = 1, \dots, m\}$.

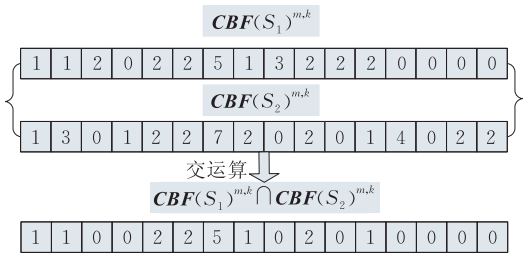


图 4 计数布鲁姆过滤器交运算

定义 6(同源计数布鲁姆过滤器的减运算). 对于两同源计数布鲁姆过滤器 $CBF(S_1)^{m,k}$ 与 $CBF(S_2)^{m,k}$,其减运算:

$$CBF(S_1)^{m,k} \setminus CBF(S_2)^{m,k} = CBF(S_1)^{m,k} \cap \overline{CBF(S_2)^{m,k}} = \{C_i | C_i = \min(CBF(S_1)_i^{m,k}, (CBF(U)_i^{m,k} - CBF(S_2)_i^{m,k})), i = 1, \dots, m\}.$$

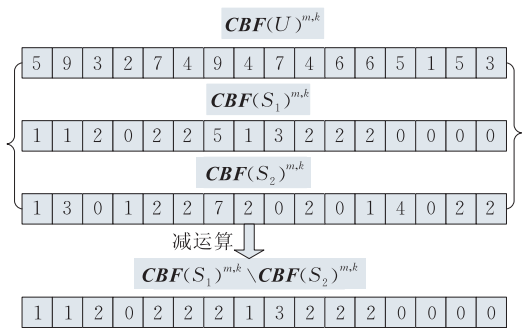


图 5 计数布鲁姆过滤器减运算

定义 7(同源计数布鲁姆过滤器的并运算). 两同源计数布鲁姆过滤器 $CBF(S_1)^{m,k}$ 与 $CBF(S_2)^{m,k}$ 的并运算如下:

$$\begin{aligned} \mathbf{CBF}(S_1)^{m,k} \cup \mathbf{CBF}(S_2)^{m,k} &= \{C_i \mid C_i = \mathbf{CBF}(S_1)_i^{m,k} + \\ &(\mathbf{CBF}(S_2)^{m,k} \cap \overline{\mathbf{CBF}(S_1)^{m,k}})_i\} = \{C_i \mid C_i = \\ &\mathbf{CBF}(S_1)_i^{m,k} + \min(\mathbf{CBF}(S_2)_i^{m,k}, (\mathbf{CBF}(U)_i^{m,k} - \\ &\mathbf{CBF}(S_1)_i^{m,k}))\} = \{C_i \mid C_i = \mathbf{CBF}(S_2)_i^{m,k} + \\ &(\mathbf{CBF}(S_1)^{m,k} \cap \overline{\mathbf{CBF}(S_2)^{m,k}})_i\} = \{C_i \mid C_i = \mathbf{CBF}(S_2)_i^{m,k} + \\ &\min(\mathbf{CBF}(S_1)_i^{m,k}, (\mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S_2)_i^{m,k}))\}. \end{aligned}$$

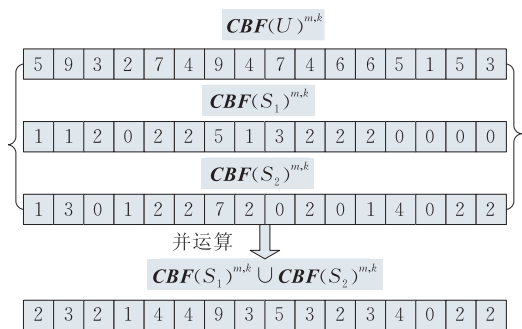


图6 计数布鲁姆过滤器并运算

定义8(同源计数布鲁姆过滤器的异或运算). 对于两同源计数布鲁姆过滤器 $\mathbf{CBF}(S_1)^{m,k}$ 与 $\mathbf{CBF}(S_2)^{m,k}$, 其异或运算:

$$\begin{aligned} \mathbf{CBF}(S_1)^{m,k} \oplus \mathbf{CBF}(S_2)^{m,k} &= (\mathbf{CBF}(S_1)^{m,k} \cap \\ &\overline{\mathbf{CBF}(S_2)^{m,k}}) \cup (\overline{\mathbf{CBF}(S_1)^{m,k}} \cap \mathbf{CBF}(S_2)^{m,k}) = \\ &(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k}) \cup (\mathbf{CBF}(S_2)^{m,k} \setminus \\ &\mathbf{CBF}(S_1)^{m,k}) = \{C_i \mid C_i = \min(\mathbf{CBF}(S_1)_i^{m,k}, \\ &(\mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S_2)_i^{m,k})) + \min(\mathbf{CBF}(S_2)_i^{m,k}, \\ &(\mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S_1)_i^{m,k})), i=1, \dots, m\}. \end{aligned}$$

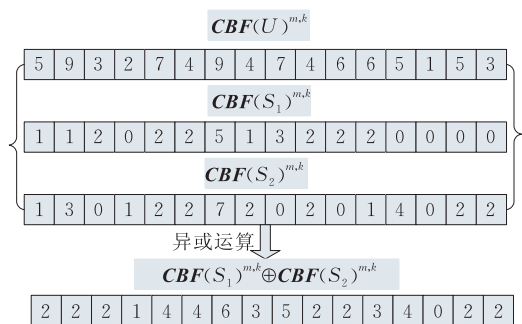


图7 计数布鲁姆过滤器异或运算

4 计数布鲁姆过滤器代数运算和集合运算的关系

定义9(同源计数布鲁姆过滤器的大小关系). 两同源计数布鲁姆过滤器 $\mathbf{CBF}(S_1)^{m,k}$ 、 $\mathbf{CBF}(S_2)^{m,k}$, 若 $\forall i \in \{1, \dots, m\}$, 都有 $\mathbf{CBF}(S_1)_i^{m,k} \geq \mathbf{CBF}(S_2)_i^{m,k}$, 则称 $\mathbf{CBF}(S_1)^{m,k}$ 大于等于 $\mathbf{CBF}(S_2)^{m,k}$, 记为 $\mathbf{CBF}(S_1)^{m,k} \geq \mathbf{CBF}(S_2)^{m,k}$; 若 $\forall i \in \{1, \dots, m\}$, 都有

$\mathbf{CBF}(S_1)_i^{m,k} \leq \mathbf{CBF}(S_2)_i^{m,k}$, 则称 $\mathbf{CBF}(S_1)^{m,k}$ 小于等于 $\mathbf{CBF}(S_2)^{m,k}$, 记为 $\mathbf{CBF}(S_1)^{m,k} \leq \mathbf{CBF}(S_2)^{m,k}$; 若 $\forall i \in \{1, \dots, m\}$, 都有 $\mathbf{CBF}(S_1)_i^{m,k} = \mathbf{CBF}(S_2)_i^{m,k}$, 则称 $\mathbf{CBF}(S_1)^{m,k}$ 等于 $\mathbf{CBF}(S_2)^{m,k}$, 记为 $\mathbf{CBF}(S_1)^{m,k} = \mathbf{CBF}(S_2)^{m,k}$.

定理1. 对于全集中的任意子集 S_1, S_2 , 其同源计数布鲁姆过滤器表示分别为 $\mathbf{CBF}(S_1)^{m,k}$ 、 $\mathbf{CBF}(S_2)^{m,k}$, 若 $S_1 \subseteq S_2$, 则有 $\mathbf{CBF}(S_1)^{m,k} \leq \mathbf{CBF}(S_2)^{m,k}$.

证明. 不失一般性, 假设 $(S_1 \cap S_2)$ 中元素先同时插入 $\mathbf{CBF}(S_1)^{m,k}$ 、 $\mathbf{CBF}(S_2)^{m,k}$ 中, 此时 $\mathbf{CBF}(S_1)_i^{m,k} = \mathbf{CBF}(S_2)_i^{m,k}$, $i=1, \dots, m$, 而后差集 $(S_2 \setminus S_1)$ 中元素插入 $\mathbf{CBF}(S_2)^{m,k}$ 中, 因 $S_1 \subseteq S_2$, $S_2 \setminus S_1$ 为空或不空, 则 $\mathbf{CBF}(S_1)_i^{m,k} \leq \mathbf{CBF}(S_2)_i^{m,k}$, $i=1, \dots, m$, 即 $\mathbf{CBF}(S_1)^{m,k} \leq \mathbf{CBF}(S_2)^{m,k}$. 定理得证. 证毕.

4.1 计数布鲁姆过滤器补运算与补集的计数布鲁姆过滤器表示

定理2. 对于全集 U 中的任意子集 S 及补集 $\bar{S}$, $\mathbf{CBF}(S)^{m,k}$ 、 $\mathbf{CBF}(\bar{S})^{m,k}$ 分别为它们的同源计数布鲁姆过滤器表示, 则 $\mathbf{CBF}(\bar{S})^{m,k} = \overline{\mathbf{CBF}(S)^{m,k}} = \{C_i \mid C_i = \mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S)_i^{m,k}, i=1, \dots, m\}$.

证明. 由于计数布鲁姆过滤器支持元素的删除操作, 因此补集 $\bar{S}$ 的布鲁姆过滤器向量可以通过如下方法得到: 先将全集 U 中元素映射到计数布鲁姆过滤器, 得 $\mathbf{CBF}(U)^{m,k}$, 然后从 $\mathbf{CBF}(U)^{m,k}$ 中逐个删除 S 中元素, 所得计数布鲁姆过滤器为 $\mathbf{CBF}(\bar{S})^{m,k}$, 即

$$\mathbf{CBF}(\bar{S})^{m,k} = \{C_i \mid C_i = \mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S)_i^{m,k}, i=1, \dots, m\}.$$

而由同源计数布鲁姆过滤器补运算的定义:

$$\overline{\mathbf{CBF}(S)^{m,k}} = \{C_i \mid C_i = \mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S)_i^{m,k}, i=1, \dots, m\}.$$

综上, 可得

$$\mathbf{CBF}(\bar{S})^{m,k} = \overline{\mathbf{CBF}(S)^{m,k}} =$$

$$\{C_i \mid C_i = \mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S)_i^{m,k}, i=1, \dots, m\}.$$

定理得证. 证毕.

定理3. 对于全集 U 中的任意子集 S 及补集 $\bar{S}$, 令集合规模 $|U|=u$, $|S|=n_0$, $|\bar{S}|=u-n_0$, 计数布鲁姆过滤器补运算 $\overline{\mathbf{CBF}(S)^{m,k}}$ 的假阳性概率与 $\mathbf{CBF}(\bar{S})^{m,k}$ 的假阳性概率相等, 均为 $(1 - e^{-k(u-n_0)/m})^k$.

证明. 由于补集 $\bar{S}$ 的计数布鲁姆过滤器表示 $(\mathbf{CBF}(\bar{S})^{m,k})$ 其假阳性概率: $f_{p_{m,u-n_0,k}}^{\mathbf{CBF}} = (1 - e^{-k(u-n_0)/m})^k$, 且由定理2知 $\overline{\mathbf{CBF}(S)^{m,k}} = \mathbf{CBF}(\bar{S})^{m,k}$. 故 $\overline{\mathbf{CBF}(S)^{m,k}}$ 假阳性概率也等于 $(1 - e^{-k(u-n_0)/m})^k$.

定理得证.

证毕.

4.2 计数布鲁姆过滤器交运算与交集的计数布鲁姆过滤器表示

定理 4. 对于全集 U 中的任意子集 S_1, S_2 , $\mathbf{CBF}(S_1)^{m,k}, \mathbf{CBF}(S_2)^{m,k}$ 分别为它们的同源计数布鲁姆过滤器表示, 则 $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k}) \geq \mathbf{CBF}(S_1 \cap S_2)^{m,k}$.

证明. 同源计数布鲁姆过滤器交运算与交集的计数布鲁姆过滤器的关系:

设 $\mathbf{CBF}(S_1 \cap S_2)_i^{m,k} = c, 1 \leq i \leq m, c \geq 0$, 则据定理 1 有 $\mathbf{CBF}(S_1)_i^{m,k} \geq c, \mathbf{CBF}(S_2)_i^{m,k} \geq c \rightarrow \forall i (1 \leq i \leq m), \min(\mathbf{CBF}(S_1)_i^{m,k}, \mathbf{CBF}(S_2)_i^{m,k}) \geq c$ 即 $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_i \geq c \Rightarrow$ 若 $\mathbf{CBF}(S_1 \cap S_2)_i^{m,k} = c$, 则必有 $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_i \geq c, 1 \leq i \leq m$. 即

$$(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k}) \geq \mathbf{CBF}(S_1 \cap S_2)^{m,k}.$$

定理得证.

证毕.

定理 5. 对于全集 U 中的任意子集 S_1, S_2 , 令集合规模 $|S_1| = n_1, |S_2| = n_2, |S_1 \cap S_2| = n_3$, $\mathbf{CBF}(S_1)^{m,k}, \mathbf{CBF}(S_2)^{m,k}$ 分别为 S_1, S_2 的同源计数布鲁姆过滤器表示, 则两过滤器交运算 $\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k}$ 假阳性概率等于

$$(1 - e^{-kn_1/m} - e^{-kn_2/m} + e^{-k(n_1+n_2-n_3)/m})^k.$$

证明. 同源计数布鲁姆过滤器交运算后其值分析如下:

(1) $\forall x \in S_1 \cap S_2$, 则 $\forall j (1 \leq j \leq k), \mathbf{CBF}(S_1)_{h_j(x)}^{m,k} \geq 1$ 且 $\mathbf{CBF}(S_2)_{h_j(x)}^{m,k} \geq 1 \rightarrow \forall j (j \in [1, k]), \min(\mathbf{CBF}(S_1)_{h_j(x)}^{m,k}, \mathbf{CBF}(S_2)_{h_j(x)}^{m,k}) \geq 1$, 即 $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_{h_j(x)} \geq 1 \Rightarrow \forall x \in S_1 \cap S_2, (\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_{h_j(x)} \geq 1, 1 \leq j \leq k$.

(2) 若 $x \in S_1 \setminus (S_1 \cap S_2), y \in S_2 \setminus (S_1 \cap S_2)$, 存在 $j (1 \leq j \leq k)$ 和 $j' (1 \leq j' \leq k), h_j(x) = h_{j'}(y) = i$, 使 $\mathbf{CBF}(S_1)_{h_j(x)}^{m,k} \geq 1, \mathbf{CBF}(S_2)_{h_{j'}(y)}^{m,k} \geq 1 \rightarrow \min(\mathbf{CBF}(S_1)_i^{m,k}, \mathbf{CBF}(S_2)_i^{m,k}) \geq 1$, 即 $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_i \geq 1, i = h_j(x) = h_{j'}(y)$.

(3) 其它, $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_i = 0, i = 1, \dots, m$.

由以上分析可知, $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_i$ 非 0 的概率 p 等于:

$$\begin{aligned} & \left(1 - \left(1 - \frac{1}{m}\right)^{kn_3}\right) + \left(1 - \frac{1}{m}\right)^{kn_3} \left(1 - \left(1 - \frac{1}{m}\right)^{k(n_1-n_3)}\right) \\ & \left(1 - \left(1 - \frac{1}{m}\right)^{k(n_2-n_3)}\right) = 1 - e^{-kn_1/m} - e^{-kn_2/m} + \\ & e^{-k(n_1+n_2-n_3)/m}. \end{aligned}$$

故交运算 $\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k}$ 的假阳性概率为 $f p_{m, n_3, k}^{\mathbf{CBF} \cap} = p^k = (1 - e^{-kn_1/m} - e^{-kn_2/m} + e^{-k(n_1+n_2-n_3)/m})^k$. 定理得证.

证毕.

交集的计数布鲁姆过滤器中, 第 i 个计数器 $(\mathbf{CBF}(S_1 \cap S_2)_i^{m,k})$ 非 0 的概率等于 $1 - \left(1 - \frac{1}{m}\right)^{kn_3}$, 设为 p_1 , 由交运算取值分析, $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})_i$ 非 0 的概率 p 为

$$\begin{aligned} & \left(1 - \left(1 - \frac{1}{m}\right)^{kn_3}\right) + \left(1 - \frac{1}{m}\right)^{kn_3} \left(1 - \left(1 - \frac{1}{m}\right)^{k(n_1-n_3)}\right) \\ & \left(1 - \left(1 - \frac{1}{m}\right)^{k(n_2-n_3)}\right) p \geq p_1, \end{aligned}$$

从而有 $p^k \geq p_1^k$, 即 $(\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k})$ 假阳性概率大于等于 $\mathbf{CBF}(S_1 \cap S_2)^{m,k}$ 假阳性概率.

4.3 计数布鲁姆过滤器减运算与差集的计数布鲁姆过滤器表示

定理 6. 对于全集 U 中的任意子集 S_1, S_2 , $\mathbf{CBF}(S_1)^{m,k}, \mathbf{CBF}(S_2)^{m,k}$ 分别为它们的同源计数布鲁姆过滤器表示, 则 $(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k}) \geq \mathbf{CBF}(S_1 \setminus S_2)^{m,k}$.

证明. 若 $\mathbf{CBF}(S_1 \setminus S_2)_i^{m,k} = c, 1 \leq i \leq m$, 因 $(S_1 \setminus S_2) \subseteq S_1, (S_1 \setminus S_2) \subseteq \bar{S}_2$, 则据定理 1 有 $\mathbf{CBF}(S_1)_i^{m,k} \geq c$ 且 $\mathbf{CBF}(\bar{S}_2)_i^{m,k} \geq c, 1 \leq i \leq m \rightarrow \min(\mathbf{CBF}(S_1)_i^{m,k}, (\mathbf{CBF}(U)_i^{m,k} - \mathbf{CBF}(S_2)_i^{m,k})) \geq c, 1 \leq i \leq m$, 即 $(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k})_i \geq c \Rightarrow$ 若 $\mathbf{CBF}(S_1 \setminus S_2)_i^{m,k} = c$, 则必有 $(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k})_i \geq c, 1 \leq i \leq m$.

因此, $(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k}) \geq \mathbf{CBF}(S_1 \setminus S_2)^{m,k}$. 定理得证.

证毕.

定理 7. 对于全集 U 中的任意子集 S_1, S_2 , 令集合规模 $|U| = u, |S_1| = n_1, |S_2| = n_2, |S_1 \cap S_2| = n_3$, $\mathbf{CBF}(S_1)^{m,k}, \mathbf{CBF}(S_2)^{m,k}$ 分别为 S_1, S_2 的同源计数布鲁姆过滤器表示, 则两过滤器减运算 $(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k})$ 假阳性概率为

$$f p_{m, n_1-n_3, k}^{\mathbf{CBF} \setminus} = (1 - e^{-kn_1/m} - e^{-k(u-n_2)/m} + e^{-k(u-n_2+n_3)/m})^k.$$

证明. 同源计数布鲁姆过滤器减运算取值情况分析:

(1) $\forall x \in S_1 \setminus S_2$, 则 $\forall j (1 \leq j \leq k), \mathbf{CBF}(S_1)_{h_j(x)}^{m,k} \geq 1$; 且 $x \in S_1 \setminus S_2 \rightarrow x \in U \cap x \notin S_2 \rightarrow (\mathbf{CBF}(U)_{h_j(x)}^{m,k} - \mathbf{CBF}(S_2)_{h_j(x)}^{m,k}) \geq 1, j = 1, \dots, k \Rightarrow \min(\mathbf{CBF}(S_1)_{h_j(x)}^{m,k}, (\mathbf{CBF}(U)_{h_j(x)}^{m,k} - \mathbf{CBF}(S_2)_{h_j(x)}^{m,k})) \geq 1, j = 1, \dots, k$ 即 $(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k})_{h_j(x)} \geq 1 \Rightarrow \forall x \in S_1 \setminus S_2$, 有 $(\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k})_{h_j(x)} \geq 1, 1 \leq j \leq k$.

(2) 当 $x \in S_1 \cap S_2, y \in \bar{S}_1 \cap \bar{S}_2$, 存在 $j (1 \leq j \leq k)$ 和

$j'(1 \leq j' \leq k), h_j(x) = h_{j'}(y) = i$, 使 $CBF(S_1)_{h_j(x)}^{m,k} \geq 1$, $CBF(\bar{S}_2)_{h_{j'}(y)}^{m,k} \geq 1 \rightarrow \min(CBF(S_1)_{h_j(x)}^{m,k}, (CBF(U)_{h_{j'}(y)}^{m,k} - CBF(S_2)_{h_{j'}(y)}^{m,k})) \geq 1$, 即 $(CBF(S_1)_{h_j(x)}^{m,k} \setminus CBF(S_2)_{h_{j'}(y)}^{m,k})_i \geq 1$, $i = h_j(x) = h_{j'}(y)$.

(3) 其它, $(CBF(S_1)_{h_j(x)}^{m,k} \setminus CBF(S_2)_{h_{j'}(y)}^{m,k})_i = 0, i \in [1, m]$.

综上, $(CBF(S_1)_{h_j(x)}^{m,k} \setminus CBF(S_2)_{h_{j'}(y)}^{m,k})_i$ 非 0 的概率:

$$p = \left(1 - \left(1 - \frac{1}{m}\right)^{k(n_1 - n_3)}\right) + \left(1 - \frac{1}{m}\right)^{k(n_1 - n_3)} \cdot \left(1 - \left(1 - \frac{1}{m}\right)^{kn_3}\right) \cdot \left(1 - \left(1 - \frac{1}{m}\right)^{k(u - n_1 - n_2 + n_3)}\right) = 1 - e^{-kn_1/m} - e^{-k(u - n_2)/m} + e^{-k(u - n_2 + n_3)/m},$$

故 $CBF(S_1)_{h_j(x)}^{m,k} \setminus CBF(S_2)_{h_{j'}(y)}^{m,k}$ 的假阳性概率为

$$f\hat{p}_{m, n_1 - n_3, k}^{CBF \setminus} = p^k = (1 - e^{-kn_1/m} - e^{-k(u - n_2)/m} + e^{-k(u - n_2 + n_3)/m})^k.$$

定理得证. 证毕.

差集的计数布鲁姆过滤器中, 第 i 个计数器 $(CBF(S_1 \setminus S_2)_i^{m,k})$ 非 0 的概率等于 $1 - \left(1 - \frac{1}{m}\right)^{k(n_1 - n_3)}$,

设为 p_1 , 而 $(CBF(S_1)_{h_j(x)}^{m,k} \setminus CBF(S_2)_{h_{j'}(y)}^{m,k})_i$ 非 0 的概率 p 为 $\left(1 - \left(1 - \frac{1}{m}\right)^{k(n_1 - n_3)}\right) + \left(1 - \frac{1}{m}\right)^{k(n_1 - n_3)} \cdot \left(1 - \left(1 - \frac{1}{m}\right)^{kn_3}\right) \cdot \left(1 - \left(1 - \frac{1}{m}\right)^{k(u - n_1 - n_2 + n_3)}\right)$, $p \geq p_1$, 从而 $p^k \geq p_1^k$, 即 $(CBF(S_1)_{h_j(x)}^{m,k} \setminus CBF(S_2)_{h_{j'}(y)}^{m,k})$ 的假阳性概率大于等于 $(CBF(S_1 \setminus S_2)_i^{m,k})$ 假阳性概率.

4.4 计数布鲁姆过滤器并运算与并集的计数布鲁姆过滤器表示

定理 8. 对于全集中的任意子集 S_1, S_2 , 其同源计数布鲁姆过滤器表示分别为 $CBF(S_1)_{h_j(x)}^{m,k}, CBF(S_2)_{h_{j'}(y)}^{m,k}$, 若 $S_1 \cap S_2 = \emptyset$, 则有

$$CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k} = CBF(S_1 \cup S_2)_{h_j(x)}^{m,k} = \{C_i | C_i = CBF(S_1)_{h_j(x)}^{m,k} + CBF(S_2)_{h_{j'}(y)}^{m,k}, i = 1, \dots, m\}.$$

证明. 步 1, 计算 $CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k}$.

由同源计数布鲁姆过滤器并运算的定义得 $CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k} = \{C_i | C_i = CBF(S_1)_{h_j(x)}^{m,k} + \min(CBF(S_2)_{h_{j'}(y)}^{m,k}, (CBF(U)_{h_j(x)}^{m,k} - CBF(S_1)_{h_j(x)}^{m,k}))\}$, $i = 1, \dots, m$ (1)

已知 $S_1 \cap S_2 = \emptyset$, 得 $S_2 \subseteq \bar{S}_1$, 由定理 1 及定理 2 得 $CBF(S_2)_{h_{j'}(y)}^{m,k} \leq \overline{CBF(S_1)_{h_j(x)}^{m,k}}, i = 1, \dots, m$. 再由同源计数布鲁姆过滤器补运算的定义, 得

$$CBF(S_2)_{h_{j'}(y)}^{m,k} \leq (CBF(U)_{h_j(x)}^{m,k} - CBF(S_1)_{h_j(x)}^{m,k}), i = 1, \dots, m \quad (2)$$

综合式(1)、(2), 得

$$CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k} = \{C_i | C_i = CBF(S_1)_{h_j(x)}^{m,k} + CBF(S_2)_{h_{j'}(y)}^{m,k}, i = 1, \dots, m\} \quad (3)$$

步 2, 计算 $CBF(S_1 \cup S_2)_{h_j(x)}^{m,k}$.

因 $S_1 \cap S_2 = \emptyset$, 故并集的布鲁姆过滤器向量可以通过如下方法得到:

先后将 S_1, S_2 中元素插入同一个布鲁姆过滤器中. 即有

$$CBF(S_1 \cup S_2)_{h_j(x)}^{m,k} = \{C_i | C_i = CBF(S_1)_{h_j(x)}^{m,k} + CBF(S_2)_{h_{j'}(y)}^{m,k}, i = 1, \dots, m\} \quad (4)$$

由式(3)、(4)得

$$CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k} = CBF(S_1 \cup S_2)_{h_j(x)}^{m,k} = \{C_i | C_i = CBF(S_1)_{h_j(x)}^{m,k} + CBF(S_2)_{h_{j'}(y)}^{m,k}, i = 1, \dots, m\}.$$

定理得证. 证毕.

定理 9. 对于全集 U 中的任意子集 S_1, S_2 , $CBF(S_1)_{h_j(x)}^{m,k}, CBF(S_2)_{h_{j'}(y)}^{m,k}$ 分别为它们的同源计数布鲁姆过滤器表示, 则 $(CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k}) \geq CBF(S_1 \cup S_2)_{h_j(x)}^{m,k}$.

证明. 由同源计数布鲁姆过滤器并运算的定义, 有 $\forall i \in [1, m], (CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k})_i = CBF(S_1)_{h_j(x)}^{m,k} + \min(CBF(S_2)_{h_{j'}(y)}^{m,k}, (CBF(U)_{h_j(x)}^{m,k} - CBF(S_1)_{h_j(x)}^{m,k}))$ 由补运算的定义及定理 4 得 $\min(CBF(S_2)_{h_{j'}(y)}^{m,k}, (CBF(U)_{h_j(x)}^{m,k} - CBF(S_1)_{h_j(x)}^{m,k})) \geq CBF(S_2 \cap \bar{S}_1)_{h_{j'}(y)}^{m,k}, i = 1, \dots, m. \rightarrow (CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k})_i \geq CBF(S_1)_{h_j(x)}^{m,k} + CBF(S_2 \cap \bar{S}_1)_{h_{j'}(y)}^{m,k}$ 又因 $S_1 \cap (S_2 \cap \bar{S}_1) = \emptyset$, 由定理 8 得

$$CBF(S_1)_{h_j(x)}^{m,k} + CBF(S_2 \cap \bar{S}_1)_{h_{j'}(y)}^{m,k} = CBF(S_1 \cup (S_2 \cap \bar{S}_1))_{h_j(x)}^{m,k} = CBF(S_1 \cup S_2)_{h_j(x)}^{m,k}, i = 1, \dots, m.$$

定理得证. 证毕.

定理 10. 对于全集中的任意子集 S_1, S_2 以及它们的并集 $S_1 \cup S_2$, 其同源计数布鲁姆过滤器表示分别为 $CBF(S_1)_{h_j(x)}^{m,k}, CBF(S_2)_{h_{j'}(y)}^{m,k}, CBF(S_1 \cup S_2)_{h_j(x)}^{m,k}$, 令集合规模 $|S_1| = n_1, |S_2| = n_2, |S_1 \cap S_2| = n_3$, $CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k}$ 与 $CBF(S_1 \cup S_2)_{h_j(x)}^{m,k}$ 的假阳性概率相同

$$f\hat{p}_{m, n_1 + n_2 - n_3, k}^{CBF \cup} = (1 - e^{-(n_1 + n_2 - n_3)/m})^k.$$

证明. 对于两计数布鲁姆过滤器, 如果其计数器向量中非 0 计数器的位置一一对应, 则其假阳性概率相同.

(1) $\forall x \in S_1$ 或者 $x \in S_2 \rightarrow CBF(S_1)_{h_j(x)}^{m,k} \geq 1$ 或 $CBF(S_2)_{h_{j'}(y)}^{m,k} \geq 1, j = 1, \dots, k$. 由定义 7, 得 $(CBF(S_1)_{h_j(x)}^{m,k} \cup CBF(S_2)_{h_{j'}(y)}^{m,k})_{h_j(x)} \geq 1, j = 1, \dots, k$; 又 $\forall x \in S_1$ 或者 $x \in$

$S_2 \rightarrow x \in S_1 \cup S_2 \rightarrow CBF(S_1 \cup S_2)_{h_j(x)}^{m,k} \geq 1, j=1, \dots, k$, 从而有 $\forall x \in S_1$ 或 $x \in S_2, (CBF(S_1)^{m,k} \cup CBF(S_2)^{m,k})_{h_j(x)} \geq 1 \Rightarrow CBF(S_1 \cup S_2)_{h_j(x)}^{m,k} \geq 1$.

(2) $\forall x \in S_1 \cup S_2 \rightarrow CBF(S_1 \cup S_2)_{h_j(x)}^{m,k} \geq 1, j=1, \dots, k$; 又 $\forall x \in S_1 \cup S_2 \rightarrow \forall x \in S_1$ 或 $\forall x \in S_2 \rightarrow CBF(S_1)_{h_j(x)}^{m,k} \geq 1$ 或 $CBF(S_2)_{h_j(x)}^{m,k} \geq 1, j=1, \dots, k$. 由定义 7 可得 $(CBF(S_1)^{m,k} \cup CBF(S_2)^{m,k})_{h_j(x)} \geq 1$. 从而有 $\forall x \in S_1 \cup S_2, CBF(S_1 \cup S_2)_{h_j(x)}^{m,k} \geq 1 \Rightarrow (CBF(S_1)^{m,k} \cup CBF(S_2)^{m,k})_{h_j(x)} \geq 1$. 由(1)、(2)可知: $CBF(S_1)^{m,k} \cup CBF(S_2)^{m,k}$ 与 $CBF(S_1 \cup S_2)^{m,k}$ 非 0 计数器的位置一一对应, 从而其假阳性概率相同. 而 $CBF(S_1 \cup S_2)^{m,k}$ 的假阳性概率为 $(1 - e^{-k(n_1+n_2-n_3)/m})^k$, 从而 $CBF(S_1)^{m,k} \cup CBF(S_2)^{m,k}$ 的假阳性概率也为 $(1 - e^{-k(n_1+n_2-n_3)/m})^k$. 定理得证. 证毕.

4.5 计数布鲁姆过滤器异或运算与对称差的计数布鲁姆过滤器表示

定理 11. 对于全集 U 中的任意子集 S_1, S_2 , $CBF(S_1)^{m,k}, CBF(S_2)^{m,k}$ 分别为它们的同源计数布鲁姆过滤器表示, 则 $(CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k}) \geq CBF(S_1 \oplus S_2)^{m,k}$.

证明. 同源计数布鲁姆过滤器异或运算与对称差的计数布鲁姆过滤器的关系如下:

$$\begin{aligned} & \text{设 } CBF(S_1 \oplus S_2)_i^{m,k} = c, i \in [1, m], \text{ 则} \\ & (CBF(S_1 \setminus S_2)_i^{m,k} + CBF(S_2 \setminus S_1)_i^{m,k}) = c \\ & \rightarrow (CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k})_i \\ & = ((CBF(S_1)^{m,k} \setminus CBF(S_2)^{m,k}) \cup \\ & \quad (CBF(S_2)^{m,k} \setminus CBF(S_1)^{m,k}))_i \\ & = (CBF(S_1)^{m,k} \setminus CBF(S_2)^{m,k})_i + \\ & \quad (CBF(S_2)^{m,k} \setminus CBF(S_1)^{m,k})_i \\ & \geq CBF(S_1 \setminus S_2)_i^{m,k} + CBF(S_2 \setminus S_1)_i^{m,k} \geq c \\ & \Rightarrow (CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k}) \\ & \geq CBF(S_1 \oplus S_2)^{m,k}. \end{aligned}$$

定理 12. 对于全集中的任意子集 S_1, S_2 , 其同源计数布鲁姆过滤器表示分别为 $CBF(S_1)^{m,k}, CBF(S_2)^{m,k}$, 则 $CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k}$ 的假阳性概率

$$f p_{m, n_1+n_2-2n_3, k}^{CBF \oplus} = (1 - e^{-k(n_1+n_2-n_3)/m} - e^{-k(u-n_3)/m} + e^{-ku/m})^k.$$

证明. 同源计数布鲁姆过滤器异或运算后任意计数器可能取值分析如下:

(1) $\forall x \in S_1 \oplus S_2$, 则 $CBF(S_1 \oplus S_2)_{h_j(x)}^{m,k} \geq 1, 1 \leq j \leq k$; $\forall x \in S_1 \oplus S_2 \rightarrow x \in S_1 \setminus S_2$ 或 $x \in S_2 \setminus S_1 \rightarrow CBF(S_1 \setminus S_2)_{h_j(x)}^{m,k} \geq 1$ 或者 $CBF(S_2 \setminus S_1)_{h_j(x)}^{m,k} \geq 1, j=$

$1, \dots, k$, 由定理 8 可得

$$\begin{aligned} & (CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k})_{h_j(x)} \\ & = ((CBF(S_1)^{m,k} \setminus CBF(S_2)^{m,k}) \cup \\ & \quad (CBF(S_2)^{m,k} \setminus CBF(S_1)^{m,k}))_{h_j(x)} \\ & = \min(CBF(S_1)_{h_j(x)}^{m,k}, CBF(U)_{h_j(x)}^{m,k} - CBF(S_2)_{h_j(x)}^{m,k}) + \\ & \quad \min(CBF(S_2)_{h_j(x)}^{m,k}, CBF(U)_{h_j(x)}^{m,k} - CBF(S_1)_{h_j(x)}^{m,k}) \\ & \geq CBF(S_1 \setminus S_2)_{h_j(x)}^{m,k} + CBF(S_2 \setminus S_1)_{h_j(x)}^{m,k} \geq 1, \\ & \quad j=1, \dots, k. \\ & \Rightarrow \forall x \in S_1 \oplus S_2, (CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k})_{h_j(x)} \geq 1, \\ & \quad 1 \leq j \leq k. \end{aligned}$$

(2) 当 $x \in S_1 \cap S_2, y \in \bar{S}_1 \cap \bar{S}_2$, 存在 $j (1 \leq j \leq k)$ 和 $j' (1 \leq j' \leq k)$, $h_j(x) = h_{j'}(y) = i$, 使得 $\min(CBF(S_1)_{h_j(x)}^{m,k}, (CBF(U)_{h_{j'}(y)}^{m,k} - CBF(S_2)_{h_{j'}(y)}^{m,k})) \geq 1$ 或 $\min(CBF(S_2)_{h_j(x)}^{m,k}, (CBF(U)_{h_{j'}(y)}^{m,k} - CBF(S_1)_{h_{j'}(y)}^{m,k})) \geq 1$, 即 $(CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k})_i \geq 1$.

(3) 其它, $(CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k})_i = 0, i = 1, \dots, m$.

由以上分析可得, $(CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k})_i$ 非 0 的概率为

$$\begin{aligned} p & = \left(1 - \left(1 - \frac{1}{m}\right)^{k(n_1+n_2-2n_3)}\right) + \left(1 - \frac{1}{m}\right)^{k(n_1+n_2-2n_3)} \cdot \\ & \quad \left(1 - \left(1 - \frac{1}{m}\right)^{kn_3}\right) \cdot \left(1 - \left(1 - \frac{1}{m}\right)^{k(u-n_1-n_2+n_3)}\right) \\ & = 1 - e^{-k(n_1+n_2-n_3)/m} - e^{-k(u-n_3)/m} + e^{-ku/m}. \end{aligned}$$

故异或运算 $CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k}$ 的查询假阳性概率为 $(1 - e^{-k(n_1+n_2-n_3)/m} - e^{-k(u-n_3)/m} + e^{-ku/m})^k$. 定理得证. 证毕.

$CBF(S_1 \oplus S_2)^{m,k}$ 非 0 的概率等于 $1 - \left(1 - \frac{1}{m}\right)^{k(n_1+n_2-2n_3)}$, 设为 p_1 , 由上述异或运算取值分析可知: $(CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k})_i$ 非 0 的概率 $p \geq p_1$, 从而 $p^k \geq p_1^k$, 即 $CBF(S_1)^{m,k} \oplus CBF(S_2)^{m,k}$ 假阳性概率大于等于 $CBF(S_1 \oplus S_2)^{m,k}$ 的假阳性概率.

定理 13. 对于全集中的任意子集 S_1, S_2 , 其同源计数布鲁姆过滤器表示分别为 $CBF(S_1)^{m,k}, CBF(S_2)^{m,k}$, 则同源计数布鲁姆过滤器进行补、交、减、并及异或运算后, 分别支持补集、交集、差集、并集及对称差集合元素的删除操作, 假阴性概率均等于 0.

证明. 为便于叙述, 计数布鲁姆过滤器代数运算用 $CBF(S_1)^{m,k} X CBF(S_2)^{m,k} (X \in \{\cup, \cap, \setminus, \oplus\})$ 表示, 其中 $\overline{CBF(S_2)^{m,k}} = CBF(U)^{m,k} \setminus CBF(S_2)^{m,k}$; 集合代数运算后的计数布鲁姆过滤器用 $CBF(S_1 X S_2)^{m,k} (X \in \{\cup, \cap, \setminus, \oplus\})$ 表示, 其中补集的计数布鲁姆

过滤器 $CBF(\overline{S_2})^{m,k} = CBF(U \setminus S_2)^{m,k}$.

(1) 计数布鲁姆过滤器补运算: 由定理 2, 知 $CBF(S)^{m,k} = CBF(\overline{S})^{m,k}$, 而 $CBF(\overline{S})^{m,k}$ 不存在假阴性, 且能支持补集 $\overline{S}$ 元素的删除操作, 故 $CBF(S)^{m,k}$ 假阴性概率等于 0, 亦能支持补集 $\overline{S}$ 元素的删除操作.

(2) 计数布鲁姆过滤器交、减、并、异或运算: 计数布鲁姆过滤器只有在删除元素时计数器本该非 0 而不正确地置为 0 才会出现假阴性, 由定理 4、定理 6、定理 9、定理 11 知 $(CBF(S_1)^{m,k} X CBF(S_2)^{m,k}) \geq CBF(S_1 X S_2)^{m,k}$ ($X \in \{\cup, \cap, \setminus, \oplus\}$), 从而在 $CBF(S_1)^{m,k} X CBF(S_2)^{m,k}$ 上可完成集合 $(S_1 X S_2)$ 中元素的删除操作, 且 $CBF(S_1)^{m,k} X CBF(S_2)^{m,k}$ 的假阴性小于等于 $CBF(S_1 X S_2)^{m,k}$, 又 $CBF(S_1 X S_2)^{m,k}$ 假阴性概率等于 0, 则 $CBF(S_1)^{m,k} X CBF(S_2)^{m,k}$ 假阴性概率只能等于 0.

综上, 同源计数布鲁姆过滤器进行补、交、减、并及异或运算后, 分别支持补集、交集、差集、并集及对称差集合元素的删除操作, 假阴性概率均等于 0. 定理得证. 证毕.

5 性能分析与模拟实验

5.1 计数布鲁姆过滤器代数运算与集合代数运算的一致性

首先进行仿真实验验证 $CBF(S_1)^{m,k} X CBF(S_2)^{m,k}$ 与 $CBF(S_1 X S_2)^{m,k}$ 之间的一致性. 实验中采用英文单词或词组作集合的数据元素, 并随机产生. 全集规模 u 等于 12 000. 由全集再随机产生子集 S_1 、 S_2 和

$S_1 \cap S_2$, 两子集规模均为 6000, 交集 $S_1 \cap S_2$ 的规模为 2000. CBF 计数器位数 $r=4$. 因为 BF 或 CBF 经常使用硬件实现, 为便于布鲁姆过滤器的硬件实现, 向量大小 m 以取值 2^x 为宜, 实验中我们取 $m = \{32768, 65536, 131072\}$. Hash 地址个数 $k = \{4, 6\}$, k 值取得不大, 是为了减少 Hash 函数的计算工作量. 根据上述设置, 单个 CBF 假阳性值在 $0.01\% \sim 10\%$ 之间变化. 采用目前应用最广泛的 Hash 算法 SHA1 算法^[17] 作为 Hash 函数, 将 SHA1 Hash 函数产生的 160 位消息摘要作为布鲁姆过滤器的 Hash 地址, 其具体作法如下: 首先计算单个 Hash 地址所必需的消息摘要位数 A ($A = \lceil \log_2 m \rceil$, m 为计数器向量的大小), 当 $\lceil A \times k / 160 \rceil = T > 1$, 即一次 SHA1 算法产生的 160 位消息摘要位串不够 k 个 Hash 地址所需的位串时, 则将集合中各字符串拼接 q 次 (q 依次取 $1, 2, \dots, T$) 作为 SHA1 Hash 函数的输入串进行 Hash, 得到 $T \times 160$ 位消息摘要位串, 多于 $A \times k$ 位的消息摘要位串则丢弃不要. 根据前述实验设置, 仅需调用一次 SHA1 算法即可产生元素相应的 k 个 Hash 地址. 表 1 中, fp 期望值直接通过本文假阳性概率公式计算而得; fp 实测值是不属于全集的 6000 个元素在相关过滤器上查询结果的统计值, 即先统计被误判的元素个数 $count$, 再将 $count$ 除以 6000, 即得 fp 实测值; fn 实测值则使用集合 S 作为测试数据集, 统计假阴性比率. 模拟实验中所有 fn 实测值均为 0, 与前面理论分析结果(定理 13)一致. 表 1 列出了部分实验结果, 假阳性实测值取 100 次实验数据的平均值, 用百分比表示.

表 1 计数布鲁姆过滤器查询假阳性实测值与期望值

$fp/\%$	m	k	$CBF(S_1)$	$CBF(S_2)$	$CBF(\overline{S_2})$	$\overline{CBF(S_2)}$	$CBF(S_{\cap})$	$CBF_{\cap}$	$CBF(S_{\setminus})$	$CBF_{\setminus}$	$CBF(S_{\cup})$	$CBF_{\cup}$	$CBF(S_{\oplus})$	$CBF_{\oplus}$
实测值	32768	4	7.2672	7.3000	7.2983	7.2983	0.2297	1.2360	2.2695	2.9975	24.667	24.667	15.175	16.936
期望值	32768	4	7.2700	7.2700	7.2700	7.2700	0.2202	1.2376	2.2273	2.9695	24.700	24.700	15.103	16.890
实测值	32768	6	8.8755	8.8230	8.7823	8.7823	0.0838	1.4637	1.9653	3.2472	35.165	35.165	20.791	24.514
期望值	32768	6	8.7797	8.7797	8.7797	8.7797	0.0831	1.4461	1.9602	3.2344	35.069	35.069	20.662	24.423
实测值	65536	4	0.8892	0.8785	0.8745	0.8745	0.0167	0.0572	0.2218	0.2677	4.3813	4.3813	2.2320	2.4233
期望值	65536	4	0.8842	0.8842	0.8842	0.8842	0.0174	0.0599	0.2202	0.2654	4.3557	4.3557	2.2273	2.4202
实测值	65536	6	0.5652	0.5662	0.5762	0.5762	0.0012	0.0205	0.0880	0.1207	4.6678	4.6678	1.9535	2.2793
期望值	65536	6	0.5701	0.5701	0.5701	0.5701	0.0022	0.0220	0.0831	0.1202	4.6514	4.6514	1.9602	2.2855
实测值	131072	4	0.0837	0.0822	0.0740	0.0740	0.0013	0.0030	0.0163	0.0178	0.4870	0.4870	0.2208	0.2328
期望值	131072	4	0.0784	0.0784	0.0784	0.0784	0.0012	0.0026	0.0174	0.0194	0.4785	0.4785	0.2202	0.2316
实测值	131072	6	0.0235	0.0187	0.0210	0.0210	5.00e-5	0.0002	0.0037	0.0040	0.2552	0.2552	0.0857	0.0958
期望值	131072	6	0.0192	0.0192	0.0192	0.0192	4.48e-5	0.0002	0.0022	0.0028	0.2456	0.2456	0.0831	0.0922

由表 1 可以看出:
(1) $CBF(S_2)^{m,k}$ 与 $CBF(\overline{S_2})^{m,k}$ 的查询假阳性实测值相等, 结果与定理 2 一致.

(2) 所有 fp 实测值均与计算得到的期望值相吻合, 验证了定理 3、定理 5、定理 7、定理 10 和定理 12.
(3) $CBF(S_1)^{m,k} \cap CBF(S_2)^{m,k}$ 假阳性值大于

$\mathbf{CBF}(S_1 \cap S_2)^{m,k}$ 假阳性值, 与定理 5 的分析是一致的. 又因 $\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k}$ 假阳性值高于 $\mathbf{CBF}(S_1 \cap S_2)^{m,k}$ 的必要前提就是 $\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k} \geq \mathbf{CBF}(S_1 \cap S_2)^{m,k}$, 从而也间接验证了定理 4 的结论是正确的.

(4) $\mathbf{CBF}(S_1)^{m,k} \setminus \mathbf{CBF}(S_2)^{m,k}$ 假阳性值大于 $\mathbf{CBF}(S_1 \setminus S_2)^{m,k}$ 假阳性值, 验证了定理 6 和定理 7.

(5) $\mathbf{CBF}(S_1)^{m,k} \cup \mathbf{CBF}(S_2)^{m,k}$ 与 $\mathbf{CBF}(S_1 \cup S_2)^{m,k}$ 的查询假阳性实测值相等, 结果与定理 10 一致.

(6) $\mathbf{CBF}(S_1)^{m,k} \oplus \mathbf{CBF}(S_2)^{m,k}$ 假阳性率大于 $\mathbf{CBF}(S_1 \oplus S_2)^{m,k}$ 假阳性率, 验证了定理 11 和定理 12 的结论.

与 $\mathbf{CBF}(S_1 X S_2)^{m,k}$ 相比较, 同源计数布鲁姆过滤器补运算和并运算后, 其假阳性概率不变, 交、减、异或运算后, 假阳性概率则稍有增加; 同源计数布鲁姆过滤器的并、交、补、减、异或运算均不会出现假阴性, 能够支持元素的删除操作, 运算后仍然符合计数布鲁姆过滤器的特征; $\mathbf{CBF}(S_1)^{m,k} X \mathbf{CBF}(S_2)^{m,k}$ 与 $\mathbf{CBF}(S_1 X S_2)^{m,k}$ 之间的一致性程度较高, 能够使用 $\mathbf{CBF}(S_1)^{m,k} X \mathbf{CBF}(S_2)^{m,k}$ 完成 $(S_1 X S_2)$ 元素的成员查询(下称代数法查询). 如代数法并集查询使用 CBF 并运算 $\mathbf{CBF}(S_1)^{m,k} \cup \mathbf{CBF}(S_2)^{m,k}$ 查询元素 x 是否属于并集 $(S_1 \cup S_2)$, 其查询过程: 先计算 x 的 k 个 Hash 值 $h_1(x), h_2(x), \dots, h_k(x)$, 再检查对应计数器 $((\mathbf{CBF}(S_1)^{m,k} \cup \mathbf{CBF}(S_2)^{m,k})_{h_j(x)}, j=1, \dots, k)$ 的值, 若全大于等于 1, 则判定元素 x 属于并集 $(S_1 \cup S_2)$, 否则, 判定 x 不属于并集. 上述结论同样适用于其它计数器向量布鲁姆过滤器, 如光谱布鲁姆过滤器和动态计数过滤器等.

5.2 CBF 代数法查询与双 CBF 法查询的性能比较

使用两个原始的计数布鲁姆过滤器 $\mathbf{CBF}(S_1)^{m,k}$, $\mathbf{CBF}(S_2)^{m,k}$ 进行 $(S_1 X S_2)$ 元素的成员查询称双 CBF(double CBFs)法查询. 如双 CBF 法查询元素 x 是否属于并集 $(S_1 \cup S_2)$ (简称双 CBF 法并集查询), 其查询过程: 分别用 $\mathbf{CBF}(S_1)^{m,k}$ 和 $\mathbf{CBF}(S_2)^{m,k}$ 查询 x 是否属于 S_1 和 S_2 . 若查询返回结果同为假, 则判定 x 不属于并集 $(S_1 \cup S_2)$, 否则, 判定 x 属于并集. 本节从假阳性、假阴性、时间复杂度、空间复杂度 4 个方面对代数法查询与双 CBF 法查询的性能进行评估比较. 设 $|U|=u, |S_1|=n_1, |S_2|=n_2, |S_1 \cap S_2|=n_3$. 实验中取 $n_1=n_2=6000, u=12000, r=4, m=\{32768, 65536, 131072\}, k=\{4, 6\}$, 此时, 单个 CBF 假阳性值在 $0.01\% \sim 10\%$ 之间变化. 令 $\alpha=n_3/n_1, \alpha$ 的取值范围: $\{1/3, 1/2, 2/3\}$, 分别对应对称差规

模较大、中等、较小的情形. fp 值与 fn 值的统计方法与 5.1 节相同. 实验结果为 100 次实验数据的平均值, 所有 fp 值与 fn 值均为百分比值.

5.2.1 交集查询

(1) 假阳性. 双 CBF 法交集查询的假阳性期望值计算公式 $fp_{m,n_1,n_2,k}^\cap = (1 - e^{-kn_1/m} - e^{-kn_2/m} + e^{-k(n_1+n_2-n_3)/m})^k$, 与 $\mathbf{CBF}(S_1)^{m,k} \cap \mathbf{CBF}(S_2)^{m,k}$ 假阳性期望值相等. 模拟实验结果见图 8, 其中 CBF_AND 为 CBF 交运算(代数法交集查询)的假阳性值, double_CBFs_AND 为双 CBF 法交集查询的假阳性值, CBF(intersection)为使用 $\mathbf{CBF}(S_1 \cap S_2)^{m,k}$ 进行查询的假阳性值. CBF_AND 等于 double_CBFs_AND, 即代数法与双 CBF 法交集查询假阳性值相同, 均大于等于 CBF(intersection)假阳性值. 当 m, k 值相同时, α 值越大, CBF_AND 越接近 CBF(intersection)的值, 如 $m=32768, k=4, \alpha$ 分别取值 $1/3, 1/2, 2/3$ 时, CBF_AND 与 CBF(intersection)的比值分别为 5.38, 2.15, 1.34.

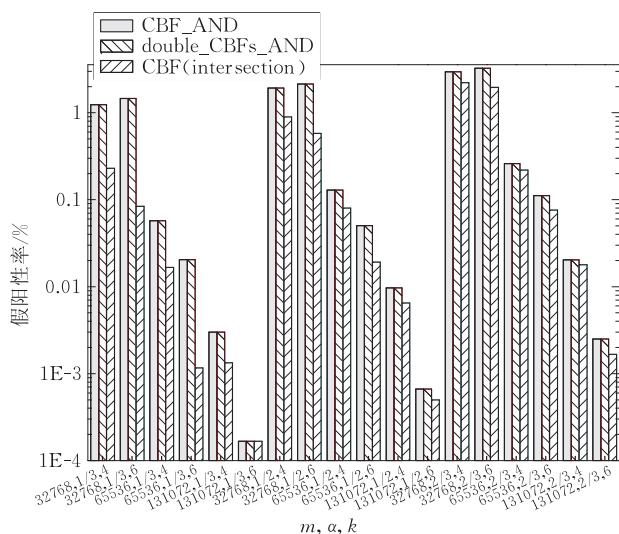


图 8 CBF 交运算、双 CBF 法交集查询与交集的计数布鲁姆过滤器假阳性

(2) 假阴性. 代数法与双 CBF 法交集查询, 假阴性率都等于 0.

(3) 空间效率. 双 CBF 法交集查询需存储两个单独的计数布鲁姆过滤器, 其占用存储空间是代数法的两倍.

(4) 时间效率. 由于实验中采用同源计数布鲁姆过滤器, 使用的 Hash 函数组相同, 因此, 双 CBF 法交集查询, 尽管需要查询两个独立的计数布鲁姆过滤器, 但 k 个 Hash 函数值只需计算一次, 其计算时间复杂度为 $O(k)$, 与代数法相同. 使用计数布鲁姆过滤器进行元素的成员查询, 除进行 Hash 函数

的计算外,还需判断元素映射的 k 个计数器是否全大于 0. 对于集合 S 中的每个元素,其所需计数器的查询次数恒定为 k ;对于不属于集合 S 的元素(若不考虑假阳性),则只需 k 个计数器中任一个为 0 即可判定其不属于 S ,其查询次数 $\leq k$. 以全集 U 作为查询测试数据集,实验中统计查询时计数器的查询次数. 设代数法查询的计数器查询次数为 T_{algebra} ,双 CBF 法查询的计数器查询次数为 $T_{\text{nonalgebra}}$,则代数法查询在计数器查询次数上的性能收益可用减幅表示,减幅计算公式为

$$(T_{\text{nonalgebra}} - T_{\text{algebra}}) / T_{\text{nonalgebra}}.$$

代数法交集查询的计数器查询次数为 $O(u + n_3) = kn_3 + k_1(u - n_3)$, $k_1 < k$; 双 CBF 法交集查询时计数器查询次数为 $O(u + n_3) = 2kn_3 + 2k_2(u - n_3)$, $k_2 < k$. 代数法交集查询的计数器查询次数少于双 CBF 法交集查询,其减幅为 $\frac{kn_3 + (2k_2 - k_1)(u - n_3)}{2kn_3 + 2k_2(u - n_3)}$. 实验结果见后面表 2. 表 2 说明,代数法交集查询的计数器查询次数比双 CBF 法交集查询减少 47%~60%.

与双 CBF 法交集查询相比,代数法交集查询假阳性概率不变,空间效率提高一倍,时间效率得到显著提高.

5.2.2 并集查询

(1) 假阳性. 双 CBF 法并集查询的假阳性期望值 $f p_{m, n_1, n_2, k}^U = (1 - e^{-kn_1/m})^k + (1 - e^{-kn_2/m})^k - (1 - e^{-kn_1/m} - e^{-kn_2/m} + e^{-k(n_1 + n_2 - n_3)/m})^k$. 根据定理 10,代数法并集查询的假阳性期望值: $f p_{m, n_1 + n_2 - n_3, k}^{CBF \cup} = (1 - e^{-k(n_1 + n_2 - n_3)/m})^k$. 模拟实验结果见图 9, 其中, CBF_OR 为 CBF 并运算(代数法并集查询)的假阳

性值, double_CBFs_OR 为双 CBF 法并集查询的假阳性值, CBF(union) 为使用 $CBF(S_1 \cup S_2)^{m, k}$ 进行查询的假阳性值. 各参数取值相同时, CBF_OR 与 CBF(union) 假阳性值相等, 但大于等于 double_CBFs_OR 假阳性值. 当 m, k 值相同时, α 值越大, double_CBFs_OR 与 CBF(union) 的比值越接近 1, 如 $m = 32768, k = 4, \alpha$ 分别取值 $1/3, 1/2, 2/3$ 时, double_CBFs_OR 与 CBF(union) 的比值分别为 0.54, 0.64, 0.77.

(2) 假阴性. 代数法与双 CBF 法查询并集元素均不存在假阴性.

(3) 空间效率. 双 CBF 法并集查询需存储两个单独的计数布鲁姆过滤器, 其占用存储空间是代数法并集查询的两倍.

(4) 时间效率. 以全集 U 作为查询测试数据集. 实验中用查询时计数器的查询次数衡量算法的查询时间复杂度. 代数法并集查询计数器查询次数为 $O(u + n_1 + n_2 - n_3) = k(n_1 + n_2 - n_3) + k_1(u - (n_1 + n_2 - n_3))$, $k_1 < k$; 双 CBF 法并集查询计数器查询次数为 $O(u + n_1 + n_2 - n_3) = k(n_1 + n_2 - n_3) + 2k_2(u - (n_1 + n_2 - n_3))$, $k_2 < k$. 代数法并集查询计数器查询次数比双 CBF 法少, 其减幅为

$$\frac{(2k_2 - k_1)(u - (n_1 + n_2 - n_3))}{k(n_1 + n_2 - n_3) + 2k_2(u - (n_1 + n_2 - n_3))}.$$

模拟实验结果见表 2. 模拟实验中减幅为 10%~18%.

与双 CBF 法并集查询相比, 代数法并集查询虽然增加了假阳性概率, 但其空间效率提高了一倍, 时间效率亦有较大程度的提高.

5.2.3 差集查询

(1) 假阳性. 双 CBF 法差集查询的假阳性期望值 $f p_{m, n_1, n_2, k}^{\setminus} = (1 - e^{-kn_1/m})^k - (1 - e^{-kn_1/m} - e^{-kn_2/m} + e^{-k(n_1 + n_2 - n_3)/m})^k$. 由定理 7, 代数法差集查询的假阳性期望值 $f p_{m, n_1 - n_3, k}^{CBF \setminus} = (1 - e^{-kn_1/m} - e^{-k(u - n_2)/m} + e^{-k(u - n_2 + n_3)/m})^k$. 模拟实验结果见图 10, 其中 CBF_SUB 为 CBF 减运算(代数法差集查询)的假阳性值, double_CBFs_SUB 为双 CBF 法差集查询的假阳性值, CBF(set differences) 为使用 $CBF(S_1 \setminus S_2)^{m, k}$ 进行查询的假阳性值. CBF_SUB 假阳性值显著小于 double_CBFs_SUB 的假阳性值, 但大于等于 CBF(set differences) 假阳性. α 值越小, CBF_SUB 假阳性值越接近 CBF(set differences) 的假阳性值, 如 $m = 32768, k = 4, \alpha$ 分别取值 $1/3, 1/2, 2/3$ 时, CBF_SUB 与 CBF(set differences) 的比值分别为 1.32, 2.13, 5.62.

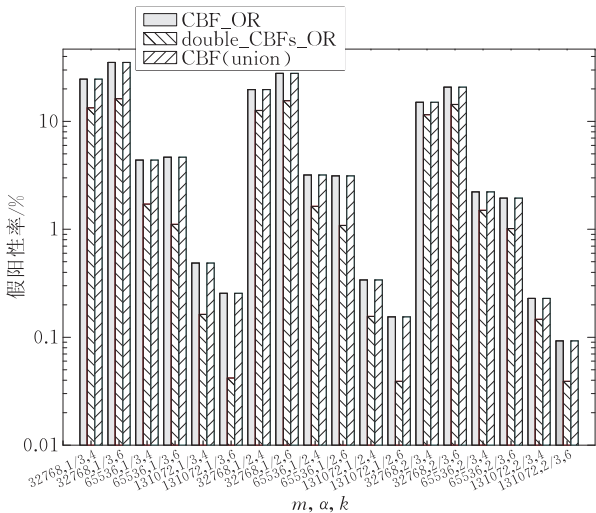


图 9 CBF 并运算、双 CBF 法并集查询与并集的计数布鲁姆过滤器假阳性

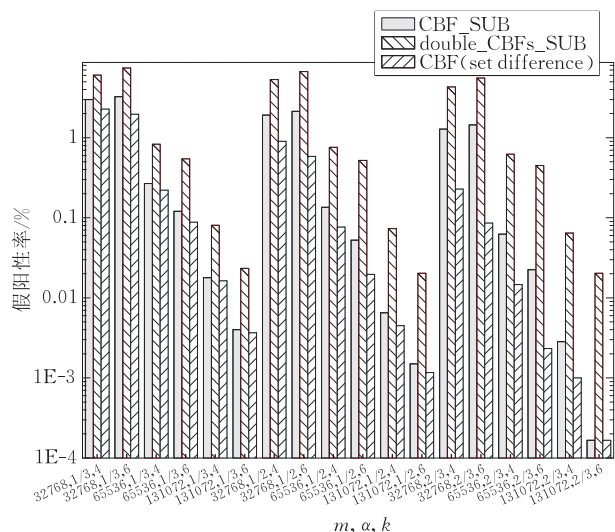


图 10 代数法、双 CBF 法差集查询与差集的
计数布鲁姆过滤器假阳性

(2) 假阴性. 双 CBF 法查询 x 是否属于差集 $S_1 \setminus S_2$ 时, 首先使用 $\mathbf{CBF}(S_1)^{m,k}$ 判断 x 是否属于 S_1 , 若是, 则继续使用 $\mathbf{CBF}(S_2)^{m,k}$ 判断 x 是否不在 S_2 中, 即判断 x 是否属于 $\bar{S}_2$. 由于在 $\mathbf{CBF}(S_2)^{m,k}$ 上进行 $\bar{S}_2$ 成员查询, 其判断规则是: 若某元素 x 用 $\mathbf{CBF}(S_2)^{m,k}$ 查询, 其不属于 S_2 , 则判定 x 属于 $\bar{S}_2$. 由于 $\mathbf{CBF}(S_2)^{m,k}$ 存在假阳性, 故用 $\mathbf{CBF}(S_2)^{m,k}$ 查询 $\bar{S}_2$ 元素会出现假阴性, 从而双 CBF 法查询差集元素时, 会出现假阴性, 其假阴性期望值为 $f n_{m,n_1,n_2,k}^- = (1 - e^{-kn_2/m})^k$, 实验统计值为 $0.02\% \sim 8.79\%$. 而代数法查询差集元素则不存在假阴性.

(3) 空间效率. 双 CBF 法差集查询需存储两个单独的计数布鲁姆过滤器, 其占用存储空间是代数法差集查询的两倍.

(4) 时间效率. 以全集 U 作为查询测试数据集, 实验中统计查询时计数器的查询次数. 代数法差集查询的计数器查询次数为 $O(u + n_1 - n_3) = k(n_1 - n_3) + k_1(u - n_1 + n_3)$, $k_1 < k$; 双 CBF 法差集查询计数器查询次数为 $O(u + n_1 - n_3) = k(n_1 - n_3) + 2kn_3 + 2k_2(u - n_1) + k_3(n_1 - n_3)$, $k_2 < k$, $k_3 < k$. 代数法差集查询在计数器查询次数上的减幅为

$$\frac{k_3(n_1 - n_3) + (2k_2 - k_1)(u - n_1) + (2k - k_1)n_3}{k(n_1 - n_3) + 2kn_3 + 2k_2(u - n_1) + k_3(n_1 - n_3)}.$$

模拟实验结果见表 2. 实验结果表明, 与双 CBF 法差集查询相比, 代数法差集查询计数器查询次数减少 $42\% \sim 65\%$.

与双 CBF 法差集查询相比, 代数法差集查询不存在假阴性, 假阳性率显著降低, 空间效率提高一倍, 时间效率得到显著提高.

5.2.4 对称差查询

(1) 假阳性:

双 CBF 法对称差查询假阳性期望值 $f p_{m,n_1,n_2,k}^\oplus = (1 - e^{-kn_1/m})^k + (1 - e^{-kn_2/m})^k - 2(1 - e^{-kn_1/m} - e^{-kn_2/m} + e^{-k(n_1+n_2-n_3)/m})^k$. 根据定理 12, 计数布鲁姆过滤器异或运算的假阳性值计算公式: $f p_{m,n_1+n_2-2n_3,k}^{\oplus} = (1 - e^{-k(n_1+n_2-n_3)/m} - e^{-k(u-n_3)/m} + e^{-ku/m})^k$.

模拟实验结果见图 11, 其中 CBF_XOR 为 CBF 异或运算(代数法对称差查询)的假阳性值, double_CBFs_XOR 为双 CBF 法对称差查询的假阳性值, CBF(symm. diff.) 为使用 $\mathbf{CBF}(S_1 \oplus S_2)^{m,k}$ 进行对称差查询的假阳性值. 当 $\alpha = 1/3$ 时, CBF_XOR 约是 CBF(symm. diff.) 的 1.05 倍 $\sim$ 1.12 倍, double_CBFs_XOR 约为 CBF(symm. diff.) 的 $1/2 \sim 4/5$; 当 $\alpha = 1/2$ 时, CBF_XOR 约是 CBF(symm. diff.) 的 1.20 倍 $\sim$ 1.67 倍, double_CBFs_XOR 约为 CBF(symm. diff.) 的 1.46 倍 $\sim$ 1.95 倍; $\alpha = 2/3$ 时, CBF_XOR 约是 CBF(symm. diff.) 假阳性值的 1.61 倍 $\sim$ 3.77 倍, double_CBFs_XOR 约是 CBF(symm. diff.) 的 3.80 倍 $\sim$ 11.76 倍. α 值越小, CBF_XOR 假阳性值越接近 CBF(symm. diff.) 的假阳性值.

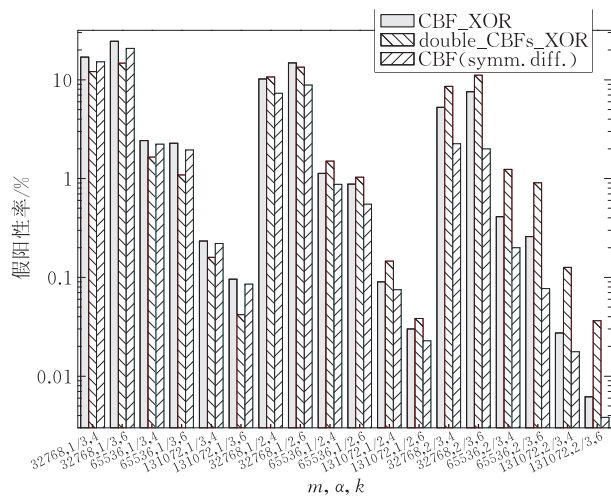


图 11 代数法、双 CBF 法对称差查询与对称差的
计数布鲁姆过滤器假阳性

(2) 假阴性. 双 CBF 法对称差查询时, 由于使用 $\mathbf{CBF}(S_1)^{m,k}$ 、 $\mathbf{CBF}(S_2)^{m,k}$ 分别查询元素是否属于 $\bar{S}_1$ 、 $\bar{S}_2$, 会出现假阴性, 故双 CBF 法查询对称差元素时, 会出现假阴性, 其假阴性期望值为 $f n_{m,n_1,n_2,k}^\oplus = (1 - e^{-kn_1/m})^k = (1 - e^{-kn_2/m})^k$, 实验统计值为 $0.02\% \sim 8.82\%$. 而代数法查询对称差元素则不存

在假阴性.

(3)空间效率. 双 CBF 法对称差查询需存储两个单独的计数布鲁姆过滤器,其占用存储空间是代数法对称差查询的两倍.

(4)时间效率. 以全集 U 作为查询测试数据集, 实验中统计查询时计数器的查询次数. 代数法对称差查询的计数器查询次数为 $O(u+n_1+n_2-2n_3)=k(n_1+n_2-2n_3)+k_1(u-n_1-n_2+2n_3), k_1 < k$; 双 CBF 法对称差查询时计数器查询次数为 $O(u+n_1+n_2-2n_3)=k(n_1+n_2)+k_2(u-n_1-n_2)+k_3(n_1+$

$n_2-2n_3), k_2 < k, k_3 < k$. 代数法对称差查询在计数器查询次数上的减幅为

$$\frac{k_3(n_1+n_2)+2(k-k_1-k_3)n_3+(k_2-k_1)(u-n_1-n_2)}{k(n_1+n_2)+k_2(u-n_1-n_2)+k_3(n_1+n_2-2n_3)}.$$

模拟实验结果见表 2. 代数法对称差查询,其计数器查询次数比双 CBF 法少,其减幅为 38%~60%.

与双 CBF 法对称差查询相比,代数法不存在假阴性,空间效率提高一倍,时间效率显著提高, α 较小时假阳性率稍有提高, α 较大时假阳性率差不多或大幅降低.

表 2 CBF 代数运算法与双 CBF 法查询计数器查询次数对比

m, α, k	并集查询次数			交集查询次数			差集查询次数			对称差查询次数			补集查询次数		
	代数法	双CBF法	减幅/%	代数法	双CBF法	减幅/%	代数法	双CBF法	减幅/%	代数法	双CBF法	减幅/%	代数法	双CBF法	减幅/%
32768,1/3,4	45094	53915	16	26387	52769	50	30339	52769	43	42200	71130	41	35570	83577	57
32768,1/2,4	43209	51914	17	28267	54799	48	28255	54799	48	38438	71145	46	35570	83577	57
32768,2/3,4	41012	49849	18	30358	56871	47	26398	56871	54	34035	71149	52	35570	83577	57
32768,1/3,6	68096	79183	14	37778	78035	52	43813	78035	44	64200	104819	39	52424	124419	58
32768,1/2,6	65259	76085	14	40603	81139	50	40604	81139	50	58528	104822	44	52424	124419	58
32768,2/3,6	61742	72865	15	43829	84401	48	37799	84401	55	51488	104842	51	52424	124419	58
65536,1/3,4	43522	51259	15	21799	46464	53	26863	46464	42	39047	65147	40	32564	80571	60
65536,1/2,4	41032	48695	16	24273	49026	50	24260	49026	51	34061	65147	48	32564	80571	60
65536,2/3,4	38373	46124	17	26869	51592	48	21786	51592	58	28744	65144	56	32564	80571	60
65536,1/3,6	64760	73592	12	28399	65357	57	36785	65357	44	57510	92632	38	46312	118315	61
65536,1/2,6	60621	69328	13	32487	69611	53	32488	69611	53	49232	92627	47	46312	118315	61
65536,2/3,6	56146	65044	14	36786	73888	50	28386	73888	62	40283	92623	57	46312	118315	61
131072,1/3,4	42688	49549	14	19714	43970	55	25301	43970	42	37369	62346	40	31171	79173	61
131072,1/2,4	39915	46747	15	22481	46777	52	22478	46777	52	31827	62349	49	31171	79173	61
131072,2/3,4	37071	43944	16	25306	49579	49	19715	49579	60	26139	62348	58	31171	79173	61
131072,1/3,6	63138	70484	10	24748	61128	60	34079	61128	44	54278	87743	38	43877	115874	62
131072,1/2,6	58504	65804	11	29370	65822	55	29370	65822	55	45012	87750	49	43877	115874	62
131072,2/3,6	53743	61115	12	34080	70502	52	24751	70502	65	35492	87745	60	43877	115874	62

5.2.5 补集查询

(1)假阳性. 双 CBF 法补集查询使用 $CBF(U)^{m,k}$ 和 $CBF(S_2)^{m,k}$ 查询元素是否属于补集 $(\bar{S}_2)$,其假阳性期望值: $fp_{m,u,n_2,k}^{\sim}=(1-e^{-ku/m})^k-(1-e^{-kn_2/m})^k$, CBF 补运算其假阳性期望值 $fp_{m,u-n_2,k}^{CBF\sim}=(1-e^{-k(u-n_2)/m})^k$. 模拟实验结果见图 12,其中 CBF_COMP 为 CBF 补运算(代数法补集查询)的假阳性值, double_CBFs_COMP 为双 CBF 法补集查询的假阳性值, CBF(complement)为使用 $CBF(\bar{S}_2)^{m,k}$ 进行补集查询的假阳性值. CBF_COMP 假阳性值等于 CBF(complement)的假阳性值,而 double_CBFs_COMP 约是 CBF(complement)的 3.78~28.11 倍. CBF_COMP、double_CBFs_COMP 、CBF(complement)与 α 值无关.

(2)假阴性. 双 CBF 法补集查询使用 $CBF(U)^{m,k}$ 和 $CBF(S_2)^{m,k}$ 进行补集元素的查询,由于用

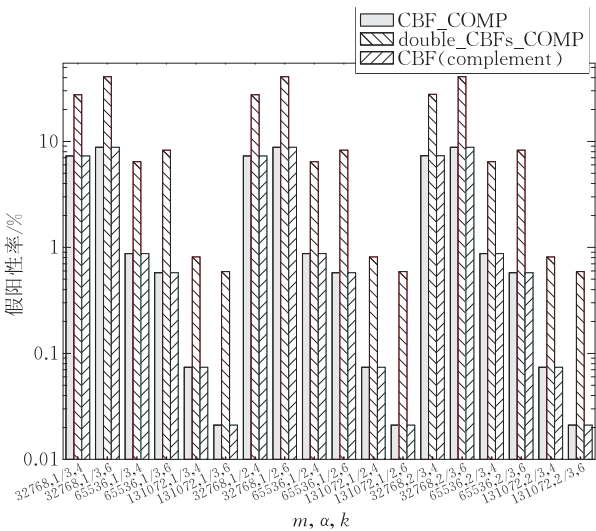


图 12 CBF 补运算、双 CBF 法补集查询与补集的计数布鲁姆过滤器假阳性

$CBF(S_2)^{m,k}$ 查询 $\bar{S}_2$ 元素会出现假阴性, 从而双 CBF 法查询补集元素时存在假阴性, 其假阴性期望值为 $f n_{m,u,n_2,k}^{\sim} = (1 - e^{-k n_2 / m})^k$, 其实验统计值为 $0.02\% \sim 8.78\%$. 而代数法补集查询不存在假阴性.

(3) 空间效率. 双 CBF 法补集查询需存储两个单独的计数布鲁姆过滤器, 其占用存储空间是代数法补集查询的两倍.

(4) 时间效率. 以全集 U 作为查询测试数据集, 实验中统计补集查询时计数器的查询次数. 代数法补集查询的计数器查询次数为 $O(u + n_2) = k(u - n_2) + k_1 n_2, k_1 < k$; 双 CBF 法补集查询时计数器查询次数为 $O(u + n_2) = ku + k_2 n_2, k_2 < k$. 代数法补集查询的计数器查询次数比双 CBF 法补集查询小, 其减幅等于 $\frac{(k - k_1 + k_2) n_2}{ku + k_2 n_2}$. 不论代数法补集查询还是双 CBF 法补集查询, 其计数器查询次数取决于 u, n_2 的值, 与交集规模 n_3 无关. 模拟实验结果见表 2. 代数法补集查询, 其计数器查询次数比双 CBF 法小, 其减幅为 $57\% \sim 62\%$.

与双 CBF 法补集查询相比, 代数法不存在假阴性, 假阳性率显著降低, 空间效率提高一倍, 时间效率得到显著提高.

由上述模拟实验与分析, 可得出结论:

(1) 对于并集查询与交集查询来说, 双 CBF 法查询与 CBF 代数法查询均不存在假阴性, 查询结果的可靠性均较高, 但是 CBF 代数法的空间效率提高一倍, 查询时间效率亦有显著提高.

(2) 对于差集查询、对称差查询与补集查询来说, 与双 CBF 法查询相比, 代数法查询的最主要优点是不存在假阴性, 另外, 代数法查询的空间效率提高一倍, 查询时间效率可显著提高.

(3) 与双 CBF 法查询的假阳性相比, 代数法交集查询假阳性概率不变, 并集查询假阳性稍有提高, 差集查询、补集查询假阳性则显著降低, 对称差查询假阳性总体上稍低.

(4) 总地说来, 对于以数据查询为主要操作的数据库或网络应用来说, 代数法查询能显著提高查询效率及查询结果的可靠性.

综合本文工作和文献[14]的工作, 可得出 CBF 代数运算及 BF 代数运算在分布式系统中的应用技巧:

(1) 当将布鲁姆过滤器并运算或交运算用于资源路由 (Resource Routing)^[5,15]、多关键字搜索 (Multi-keyword Search)^[12] 中的 And 查询、Or 查询

等分布式应用领域时, 由于 BF 并(交)运算与 CBF 并(交)运算都可以得到并(交)集的近似过滤器表示, 且查询假阳性概率一致(参见本文定理 10、定理 5 和文献[14]的定理 2、定理 4), 此时使用 BF 并(交)运算具有可以节约过滤器的向量存储空间, 代数运算更为简单的优点. 例如, 资源路由的路由表是由许多 BF 或 DBF 组成, 每个这样的过滤器保存的是通过某条链路可达的资源列表. 当节点在转发查询时, 逐个查询路由表中的 BF, 从中选择合适的链路转发. 其路由表的构造过程如下: 各结点构造本地 BF, 然后将 BF 表示的路由通告发送给邻居, 接收到其它节点的 BF 后将其与本地 BF 合并(即 BF 并运算). 资源路由时使用 BF 并运算即可以完成路由的合并, 可以节约过滤器的向量存储空间, 减少路由通告时发送给邻居的消息传输位数. 但由于 BF 不适合动态删除频繁的系统环境, 当删除操作频繁时, 此时可以使用 CBF 反映资源的删除操作, 而在路由表的构造过程中则仍然传输与 CBF 对应的用 BF 表示的路由通告.

(2) 由于 CBF 减(异或、补)运算能用于差集(对称差、补集)的成员查询, 而 BF 减(异或、补)运算不能, 当分布式应用中使用布鲁姆过滤器求解完整差集(对称差、补集)时, 只能采用 CBF 减(异或、补)运算来完成, 不能使用 BF 减(异或、补)运算. 本文第 6 节以 CBF 减运算为例讨论 CBF 代数运算应用于分布式系统的有效性和高效性.

6 CBF 减运算的应用探讨与仿真实验

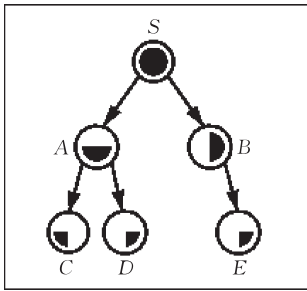
6.1 P2P 网络中的大规模文件分发系统

由于分布式系统中网络异构性和网络波动(churn)的普遍存在, 在接收相同文件内容的对等节点间, 各节点的已接收文件块往往存在显著的差异; 另外, 由于为充分利用网络带宽、提高下载速度、缩短下载时间, 需要从其它对等节点并行进行下载; 甚至存在如下可能: 由于覆盖网中网络连接的短暂性, 服务器和某对等节点间的连接断开, 这些情况下, 对等节点需要高效的集合调和(Set Reconciliation)方法^[18], 调和位于不同对等节点的已接收文件块, 最终得到所有文件块.

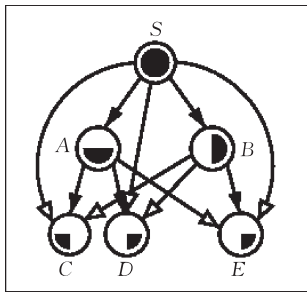
假如要将一个 800 MB 的大文件从服务器分发到诸多对等节点, 每个文件块的大小为 100 KB, 则该文件可分为 8000 个文件块. 这 8000 个文件块组成了含 8000 个元素的集合. 而且, 在现实的大规模

文件分发系统中,文件大小远超过 800 MB,最小的都是数个 GB 大小,若直接对文件块集合进行操作,会产生大量的消息数据,因此,我们使用 BF 或 CBF 对文件块集合进行精简表示。

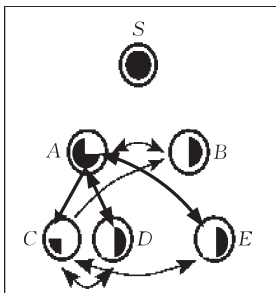
在图 13 所示的知情内容分发系统^[18]中,对等节点间在下载时相互协作,从服务器下载某些文件块(图 13(b))的同时从其它对等节点下载其它文件块(图 13(c)),以有效地提高下载速率,在知情传输的过程中发送节点仅需传送接收节点缺失的文件块,即过滤掉了重复元素,故称此内容分发过程为知情内容分发(Informed content delivery). 在知情内容分发系统的知情传输过程中,利用图 14(a)所示的集合调和的方法,发送方可找出两节点中大部分相异文件块,在节点间仅传送这些相异文件块即可,免去发送方传送共有文件块的网络开销;利用纠错码(Erasure correcting code)^[19]技术,接收方在收到发送方大部分相异编码文件块后,进行译码即能得到双方拥有文件块的总和。



(a) 初始分发树



(b) 下载



(c) 知情传输(使用集合调和的方法)

图 13 知情内容分发

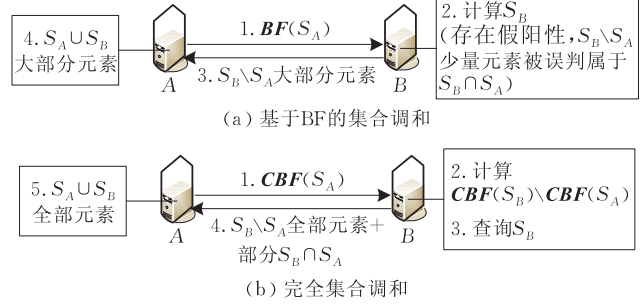


图 14 使用 BF 或 CBF 的集合调和示意图

6.2 集合调和

假定两节点 A 、 B 分别拥有文件块集合 S_A 、 S_B , 节点 A 发送 S_A 的某种表示给节点 B , 节点 B 据此计算差集 $S_B \setminus S_A$, 返回 $S_B \setminus S_A$ 中文件块给 A , 得到 $S_A \cup S_B$, 此过程称集合调和。

目前存在两大类集合调和的方法: 精确集合调和(Exact Set Reconciliation)、近似集合调和(Approximate Set Reconciliation)。精确集合调和能得到 $S_B \setminus S_A$ 的准确成员。精确集合调和最直观的方法是节点 A 直接发送 S_A 所有元素给节点 B , 设元素位串长度为 len , 此时需传输 $O(|S_A|len)$ 位消息^[18], 此法下称完整集合传输法。还有一种常用的精确集合调和的方法是特征多项式法^[20], 在应用该法前, 需将全集 U 中所有长为 len 的位串映射到有限域 $F_q (q \geq 2^{len})$ 中的元素, 在假定 $d = |S_A \setminus S_B| + |S_B \setminus S_A|$ 已知的情况下, A 需发送 $O(d \log len)$ 位数数据消息给 B ; 若 d 未知, 则必须通过 A 、 B 之间的多轮消息交换估算出 d 的合理上界, 然后根据 d 对特征多项式进行插值与因式分解, 恢复差集元素。此法所需时间复杂度为 $\Theta(d^3)$ 。文献^[20]的作者后又改进了该特征多项式法, 使其只需 $\Theta(d)$ 计算时间复杂度, 但以 A 、 B 之间更多的通信轮数为代价^[21]。这些精确集合调和的方法均存在网络通信代价高的缺点, 当 $|S_B \setminus S_A|$ 较小时尤为突出。近似集合调和则只能找出 $S_B \setminus S_A$ 的部分元素。近似集合调和在节点间传递 Hash 后的集合 ($hash(S_A)$)、BF 表示的集合 ($BF(S_A)$)^[18] 或集合的近似调和树^[22] ($ART(S_A)$) 等。这些近似集合调和的方法都存在如下可能: S_B 中元素 $y (y \in \{S_B \setminus S_A\})$ 与元素 $x (x \in S_A)$ 的 Hash 值或 BF 表示相同, 从而 B 错误地判断 $y \in S_A$, 以致不能找出 $S_B \setminus S_A$ 的所有元素。这些近似集合调和的方法本质上与双 CBF 法差集查询相同, 存在假阴性, 即原本属于 $S_B \setminus S_A$ 的元素可能被判为不属于 $S_B \setminus S_A$ 。本文中将使用 BF 的近似集合调和的方法称为基于 BF 的集合调和(图 14(a))。

6.2.1 Byers 集合调和法

知情内容分发系统的知情传输过程中使用的集合调和方法是基于 BF 的集合调和的方法. 尽管其调和过程只能得到近似差集, 但由于内容分发时被分发文件使用纠错码编码, 接收节点只需全部编码分组的足够大的一个子集就可译码得到被分发文件所有内容. 以对等节点 A、B 间的调和为例, 说明知情内容分发系统中使用的调和过程如下(下称 Byers 法):

1. 服务器 S 将拟分发文件用纠错码编码, 按初始分发树的层次分发至对等节点 A、B, 调和前各自的已接收编码分组集设为 S_A^E, S_B^E .

2. 节点 A 构造 $\mathbf{BF}(S_A^E)^{m,k}$, 并将之传送给节点 B.

3. 节点 B 使用 $\mathbf{BF}(S_A^E)^{m,k}$ 查询 S_B^E 中不属于 S_A^E 的编码分组, 并返回给节点 A. 由于 $\mathbf{BF}(S_A^E)^{m,k}$ 固有的假阳性, 少量 $\{S_B^E \setminus S_A^E\}$ 中的编码分组被误判为属于 S_A^E , 从而上述查询过程只能得到 $\{S_B^E \setminus S_A^E\}$ 中的大部分编码分组, 即近似差集 $\{S_B^E \setminus S_A^E\}$.

4. 节点 A 将节点 B 返回的近似 $\{S_B^E \setminus S_A^E\}$ 与 S_A^E 合并, 得到近似 $\{S_A^E \cup S_B^E\}$.

5. 节点 A 对近似 $\{S_A^E \cup S_B^E\}$ 进行译码, 得到完整的 $S_A \cup S_B$.

6.2.2 完全集合调和

本文的 CBF 减运算揭示了集合调和的另一选择. 如果节点 A 传送 S_A 的 CBF 表示(即 $\mathbf{CBF}(S_A)^{m,k}$)给节点 B, 节点 B 使用 CBF 减运算得到的过滤器 $\mathbf{CBF}(S_B)^{m,k} \setminus \mathbf{CBF}(S_A)^{m,k}$ 查找 $S_B \setminus S_A$ 中元素, 并返回给 A, 求解 $S_A \cup S_B$, 完成集合调和. 由于在使用 $\mathbf{CBF}(S_B)^{m,k} \setminus \mathbf{CBF}(S_A)^{m,k}$ 查询元素是否属于 $S_B \setminus S_A$ 时不存在假阴性, 但具有假阳性, 从而节点 B 找出的元素中包含 $S_B \setminus S_A$ 中全部元素, 亦包含少量 $S_A \cap S_B$ 中元素(其数量取决于 $\mathbf{CBF}(\overline{S_A})^{m,k}$ 的查询假阳性), 本文将这一方法称为完全集合调和(Complete Set Reconciliation), 参见图 14(b). 由于 CBF 用 r 位计数器代替 BF 的二进制位, 利用 CBF 减运算的完全集合调和的方法, 与使用 BF 的近似集合调和的方法相比, 前者节点 A 需传输 $O(rm)$ 位信息给节点 B, 而后者仅需 $O(m)$ 位信息. 设 $|\overline{S_A}| = |S_A| = |S_B| = n$, 当 m/n 值确定时, 为保证 $S_A \cap S_B$ 中元素被误判为 $S_B \setminus S_A$ 中元素的可能性最小, 需取 $k = \ln 2 \times (m/n)$. 例, 当 $m/n = 6$, k 取 4 时, $\mathbf{CBF}(\overline{S_A})^{m,k}$ 的 fp 值近似最小, 计算得 $fp = 0.056$, 若 $r = 4$, 利用 CBF 减运算的集合调和的方法, 节点 A 需传输给节点 B 的消息位数等于 $24n$, 由 B 返回的元素集合中包含全部 $S_B \setminus S_A$ 元素, 亦包含不到 5.6% 的 $S_A \cap$

S_B 中元素. 而使用 BF 的近似集合调和的方法仅需在节点间传送 $6n$ 位消息, 但约有不到 5.6% (取决于 $\mathbf{BF}(S_A)^{m,k}$ 的查询假阳性) 的 $S_B \setminus S_A$ 中元素被遗漏, 即不能得到完整的 $S_A \cup S_B$.

与用 BF 完成集合调和的方法比较, 使用 CBF 减运算完成集合调和可以找出 $S_B \setminus S_A$ 中所有元素, 不会遗漏, 从而得到完整的 $S_A \cup S_B$; 而且 CBF 可支持删除操作, 适用于更新频繁的系统环境, 但需比 BF 法多一次 CBF 减运算, 需要 r (如 $r = 4$) 倍于 BF 法的传输消息位数及节点在首次与服务器或对等节点建立联系时需得到 $\mathbf{CBF}(U)$.

6.3 仿真实验

6.3.1 单机环境下的模拟实验

首先考察单机环境下完全集合调和算法与 Byers 集合调和法的性能表现. 为了简化实验过程, 我们考虑最简单的场景: 用随机产生的 32 位无符号整数模拟文件块, 两对等节点从服务器 S 下载一定数量的文件块后, 停止来自服务器的下载, 转而进行集合调和, 完成整个文件所有文件块的获取过程. 表 3 中调和时间是指从 A 节点发送 BF 或 CBF 给 B 节点起, 至 A 节点获得完整文件所经历的时间. 漏检数是指 Byers 法的步 3 过程中被漏检的 $\{S_B^E \setminus S_A^E\}$ 元素个数. 冗余是指完全集合调和过程中第 4 步由 B 返回的 $S_A \cap S_B$ 元素个数.

两算法共同的参数设置如下: 根据布鲁姆过滤器假阳性概率的变化趋势, m/n 值固定的情况下, 使得布鲁姆过滤器假阳性概率最小的 k 值等于 $\ln 2 \times (m/n)$. 实验中我们选取 BF 或 CBF 向量长度 $m = 32768$, $|S_A| = |S_B| = n = 4000$, Hash 函数个数 $k = 6$, 分别考察不同交集规模下的算法调和性能. 交集规模 $|S_A^E \cap S_B^E| = |S_A \cap S_B| = \{500, 1000, 2000, 3000\}$. Byers 法其余参数设置如下: 纠错码使用里德-所罗门码, 即 $RS(nn, kk, mm)$ 码, 其中, kk 个输入符号(每个符号有 mm 个比特位)被编码成 $nn (= 2^{mm} - 1)$ 个输出符号. 实验中, 因 $2^{12} < kk < 2^{13}$, 故 mm 取值 13, $nn = 8191$, 即 kk 个源数据用 $RS(8191, kk, 13)$ 码编码成 $nn (= 8191)$ 个编码数据, 接收节点 A 和发送节点 B 在调和前拥有 $(nn + |S_A^E \cap S_B^E|)/2 (< kk)$ 个编码数据, 其中 $|S_A^E \cap S_B^E| = n_0^E$. 完全集合调和法的其它参数设置如下: 全集规模 $u = kk$, 计数器位数 $r = 4$, 其中 $|S_A \cap S_B| = n_0 = n_0^E$. 仿真实验机器配置为 Pentium dual-core 1.7 GHz, 1 GB DDR2, Windows XP.

对于表 3 中每个实验参数的组合, 随机产生

100 次数据集合,完成 100 次实验,实验结果取 100 次的平均值.表 3 中第 3 列、第 8 列分别是 Byers 法和完全集合调和法的调和时间.

表 3 单机环境完全集合调和与 Byers 法性能比较								
Byers 法的结果					完全集合调和的结果			
kk	n_0^E	调和 时间/ ms	译码 时间/ ms	漏检数/ 个	u	n_0	调和 时间/ ms	冗余/ 个
5000	3000	974	948	1655	5000	3000	19	0
6000	2000	593	570	1140	6000	2000	19	0.77
7000	1000	287	266	629	7000	1000	20	2.15
7500	500	162	143	379	7500	500	20	2.15

- 从表 3 中发现:
- (1) 当交集规模取值 3000,源数据个数为 5000 时,Byers 法的调和时间为 974 ms,而完全集合调和法的调和时间仅为 Byers 法的 1.93%,完全集合调和法的调和时间远小于 Byers 法的调和时间.
- (2) 当交集规模较小,源数据个数较大时,Byers 法的调和时间较小,如最后一行所示,但其时间值仍然是完全集合调和法的 8 倍左右.
- (3) 完全集合调和过程中第 4 步由 B 返回的 $S_A \cap S_B$ 元素个数极小,如第 9 列所示,在上述实验过程中冗余数最多不超过 6 个.这意味着由于冗余造成的传输代价可以忽略不计,在实际应用中是完全可行的.
- (4) Byers 法漏检数较大,为 379~1655,远大于单独使用基于 BF 的集合调和法时的漏检期望值 $(1 - e^{-k|S_A|/m})^k \times |S_A \setminus S_B|$.这是由于 Byers 法在使用 RS 码编码时, $(nn - kk)$ 个校验数据中有很多与 $S_B \setminus S_A$ 中数据相同,造成部分 $S_B \setminus S_A$ 元素已在 $BF(S_A^E)^{m,k}$ 中得到表示,从而导致查询时这些差集元素被认为不属于 $S_B \setminus S_A$,漏检数比期望值大得较多.被漏检的差集元素即为 RS 码要纠正的删除数据,删除数据数目越大,译码所需时间越长.漏检数大直接导致 Byers 法译码时间较长.

两算法调和时间出现上述显著差异的原因是由于 Byers 法必须使用纠删码配合基于 BF 的集合调和法,才能调和得到完整的文件,纠删码的译码过程非常耗时,为 143 ms~948 ms,译码时间占整个调和时间的 88%~97%.而完全集合调和法仅需简单的 CBF 减运算及元素查询,费时很少即能完成调和.

6.3.2 P2P 环境下的仿真实验

为进一步验证完全集合调和算法的性能,进行系统仿真实验,将完全集合调和用于知情内容分发的知情传输过程,并将之与 Byers 法进行比较.

该部分实验的参数设置:除了使用 P2P 环境外,其余参数与单机环境下的模拟实验是相同的.仿真实验使用的 P2P 结构如下:基于随机产生的物理网络建立 Chord 覆盖网^[23].Chord 底层的物理网络采用 Transit-stub 模型^[24],路由器数量为 5050,物理链路 25 168 条,链路平均延迟为 40.5 ms. Chord 网络中 P2P 结点数为 200.每个 P2P 结点连在一个随机路由器上.调和时间统计结果如表 4 所示.由表 4 结果可以看出,与 Byers 法相比,完全集合调和法同样表现出了时间性能上的优势.

表 4 P2P 环境完全集合调和与 Byers 法调和时间					
Byers 法的结果			完全集合调和的结果		
kk	n_0^E	调和时间/ms	u	n_0	调和时间/ms
5000	3000	16 018	5000	3000	15 177
6000	2000	15 705	6000	2000	15 179
7000	1000	15 436	7000	1000	15 181
7500	500	15 323	7500	500	15 187

在知情内容分发系统中,节点间使用基于 BF 的集合调和法进行知情传输,只能得到部分相异分组,需配合使用纠删码技术才能恢复完整文件.拟分发文件在传输前需使用纠删码对分组进行编码,各对等节点在接收到一定数量的编码分组后需进行译码才能得到完整文件,编码和译码过程(尤其是译码)对于较大的文件来说是一笔不小的开销.此时,若改用本文的完全集合调和法完成对等节点之间的知情传输,则不需费时的编码和译码过程,时间效率很高,且不需借助其它技术即能调和所有相异分组.

7 小结与研究展望

本文直接从 CBF 计数器向量的加、减及求最小值运算出发,探讨了 CBF 代数运算的相关性质.理论分析和实验结果表明,CBF 代数运算 $(CBF(S_1)^{m,k} X CBF(S_2)^{m,k})$ 不会出现假阴性,与 $CBF(S_1 X S_2)^{m,k}$ 相比,其假阳性概率不变或略有提高,CBF 代数运算与 $CBF(S_1 X S_2)^{m,k}$ 之间的一致性程度较高,能够支持集合并集、交集、补集、差集及对称差的成员查询;利用 CBF 代数运算得到的过滤器查询元素是否属于补集、差集及对称差,与使用两个原始的计数布鲁姆过滤器进行查询的方法相比,不存在假阴性,空间效率能提高一倍,时间效率亦能显著地得到改善.本文以 CBF 代数运算中的减运算为例,成功地将之应用于分布式文件分发系统,表现出了极为卓越的调和效率.本文工作说明:使用本文的 CBF 代数运

算,可以简化利用计数布鲁姆过滤器进行的系统设计,可用于大多数计数布鲁姆过滤器的应用场合,为优化计数布鲁姆过滤器应用提供了新的思路。

进一步探讨不同长度、不同 Hash 函数下的非同源计数布鲁姆过滤器如何进行相关代数运算,成为作者的进一步研究工作. 另一个重要的工作展望是将本文的研究成果进一步应用于实际分布式网络系统中,在各应用领域优化 CBF 代数运算的性能等。

参 考 文 献

- [1] Mitzenmacher M. Compressed Bloom filters. *IEEE/ACM Transactions on Networking*, 2002, 10(5): 604-612
- [2] Xie Kun, Wen Ji-Gang, Zhang Da-Fang, Xie Gao-Gang. Bloom filter query algorithm. *Journal of Software*, 2009, 20(1): 96-108(in Chinese)
(谢鲲, 文吉刚, 张大方, 谢高岗. 布鲁姆过滤器查询算法. *软件学报*, 2009, 20(1): 96-108)
- [3] Burton H B. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 1970, 13(7): 422-426
- [4] Fan L, Cao P, Almeida J, Broder A Z. Summary cache: A scalable wide-area Web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 2000, 8(3): 281-293
- [5] Guo D, Wu J, Chen H, Luo X. Theory and network applications of dynamic Bloom filters//*Proceedings of the INFOCOM 2006*. Barcelona, 2006: 1-12
- [6] Saar C, Yossi M. Spectral Bloom filters//*Proceedings of the ACM SIGMOD International Conference on Management of Data*. San Diego: ACM, 2003: 241-252
- [7] Aguilar-Saborit J, Muntès-Mulero V, Trancoso P et al. Dynamic count filters. *ACM SIGMOD Record*, 2006, 35(1): 26-32
- [8] Dharmapurikar S, Krishnamurthy P, Taylor D. Longest prefix matching using Bloom filters. *IEEE/ACM Transactions on Networking*, 2006, 14(2): 397-409
- [9] Dharmapurikar S, Krishnamurthy P, Sproull T, Lockwood J. Deep packet inspection using parallel Bloom filters//*Proceedings of the 11th Symposium on High Performance Interconnects*. Piscataway, NJ: IEEE Computer Society. Stanford, CA, USA, 2003: 44-51
- [10] Kirsch A, Mitzenmacher M. Distance-Sensitive Bloom filters//*Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments*. Miami, Florida, 2006: 41-50
- [11] Ficara D, Giordano S, Prociassi G, Vitucci F. Multilayer compressed counting Bloom filters//*Proceedings of the INFOCOM 2008*. Phoenix, Arizona, 2008: 311-315
- [12] Reynolds P, Vahdat A. Efficient peer-to-peer keyword searching//Endler M, Schmidt D. *Middleware 2003*. Lecture Notes in Computer Science. Berlin Heidelberg: Springer-Verlag, 2003: 21-40
- [13] Broder A, Mitzenmacher M. Network applications of Bloom filters: A survey. *Internet Mathematics*, 2005, 1(4): 485-509
- [14] Xie Kun, Zhang Da-Fang, Wen Ji-Gang, Xie Gao-Gang, You Zhi-Qiang. Algebraic operations on Bloom filters. *Acta Electronica Sinica*, 2008, 36(5): 869-874(in Chinese)
(谢鲲, 张大方, 文吉刚, 谢高岗, 尤志强. 布鲁姆过滤器代数运算探讨. *电子学报*, 2008, 36(5): 869-874)
- [15] Kumar A, Xu J, Zegura E W. Efficient and scalable query routing for unstructured peer-to-peer networks//*Proceedings of the INFOCOM 2005*. Miami, FL, 2005: 1162-1173
- [16] Bonomi F, Mitzenmacher M, Panigrahy R et al. An improved construction for counting Bloom filters//Azar Y, Erlebach T. *ESA 2006*. Lecture Notes in Computer Science. Berlin Heidelberg: Springer-Verlag, 2006: 684-695
- [17] NIST. Secure hash standard. *Federal Information Processing Standards*, FIPS-180-1, April 1995
- [18] Byers J, Considine J, Mitzenmacher M, Rost S. Informed content delivery across adaptive overlay networks. *ACM SIGCOMM Computer Communication Review*, 2002, 32(4): 47-60
- [19] Luby M, Mitzenmacher M, Shokrollahi A, Spielman D. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 2001, 47(2): 569-584
- [20] Minsky Y, Trachtenberg A, Zippel R. Set reconciliation with nearly optimal communication complexity. *IEEE Transactions on Information Theory*, 2003, 49(9): 2213-2218
- [21] Minsky Y, Trachtenberg A. Practical set reconciliation. Boston University, Boston MA: Technical Report: BU ECE 2002-01, 2002
- [22] Byers J, Considine J, Mitzenmacher M. Fast approximate reconciliation of set differences. Boston University, Boston MA: Technical Report: BU ECE 2002-19, 2002
- [23] STOICA I, MORRIS R, KARGER D, et al. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 2001, 31(4): 149-160.
- [24] Zegura E W, Calvert K L, Bhattacharjee S. How to model an internetwork//*Proceedings of the IEEE INFOCOM 1996*. San Francisco, 1996: 594-602



TIAN Xiao-Mei, born in 1971, Ph.D. candidate, associate professor. Her main research interests include peer to peer network, distributed computing, etc.

ZHANG Da-Fang, born in 1959, Ph. D., professor, Ph. D. supervisor. His main research interests include dependable systems and networks, network testing, software testing, software fault tolerance, etc.

XIE Kun, born in 1978, Ph. D., associate professor. Her main research interests include dependable systems and networks, wireless network, algorithm designing, etc.

SHI Chang-Qiong, born in 1968, Ph. D. candidate, associate professor. Her main research interests include network information security, artificial intelligence, etc.

YANG Xiao-Bo, born in 1973, Ph. D. candidate, lecturer. Her main research interests include dependable systems and networks, artificial intelligence, software designing, etc.

Background

This work is supported by the National Natural Science Foundation of China under Grant Nos. 90718008, 60703097 the National Basic Research Program(973 Program) of China under Grant No. 2007CB310702 and the Scientific Research Fund of Hunan Provincial Education Department of China under Grant No. 08D055.

A Bloom filter is an excellent data structure that can succinctly represent a data set in order to support membership queries, and effectively filter out any element that does not belong to the set. It is widely used in databases, networks and distributed systems and it has great potential for distributed applications where systems need to share information about available data. Many research communities have

been taking deep researches into Bloom filter algorithms and applications, and there may well be more Bloom filter algorithms and applications to be found. Nevertheless, there are false negatives when we check whether an element x belongs to set differences, symmetric differences or complement, using double individual Bloom filters or algebra operations on standard Bloom filters. To solve this problem, we present algebra operations on counting Bloom filters. This paper examines the consistence between algebra operations on counting Bloom filters and algebra operations on data sets, and studies the membership query performances of algebra operations on counting Bloom filters, both in theory extension and simulation experiments.