2019 级

《大数据存储与管理》课程

课 程 设 计

姓　名 陈雨轩

学　号 U201914927

班　号 计算机 1902 班

日　期 2022.04.14

目　　录

1. 总体篇幅不做明确限制；
2. 突出说明实验内容、实验过程这两部分；
3. 实验背景、实验总结应精炼、客观、准确。

一、课设目的

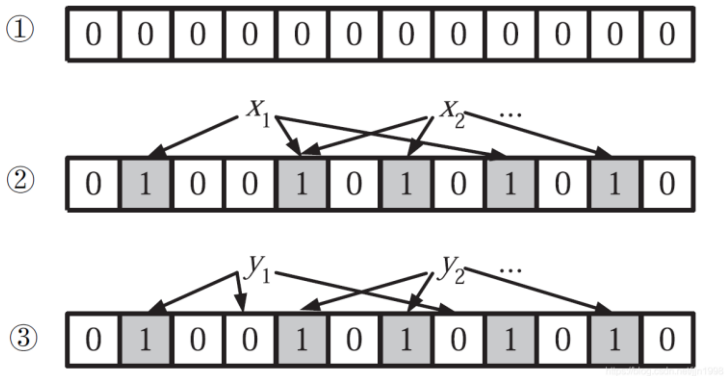
基于 Bloom Filter 的多维数据属性表示和索引

二、课设背景

Bloom filter 是一种空间效率很高的数据索引结构，它利用 bit 数组很简洁地表示一个集合，Bloom filter 的主要用来判断某个或某些元素是否属于某个集合，在判断是否属于某个集合时，有可能会把不属于这个集合的元素误认为属于这个集合（false positive）。因此，Bloom filter 不适合那些“零错误”的应用场合。而在能容忍低错误率的应用场合下，Bloom filter 可以通过极少的错误换取存储空间的极大节省。

三、课设原理

结合下图具体来看 Bloom filter 是如何通过使用位数组表示集合。



①：初始状态，此时 Bloom filter 是一个新建的包含 m 位的位数组，每一位都置为 0。

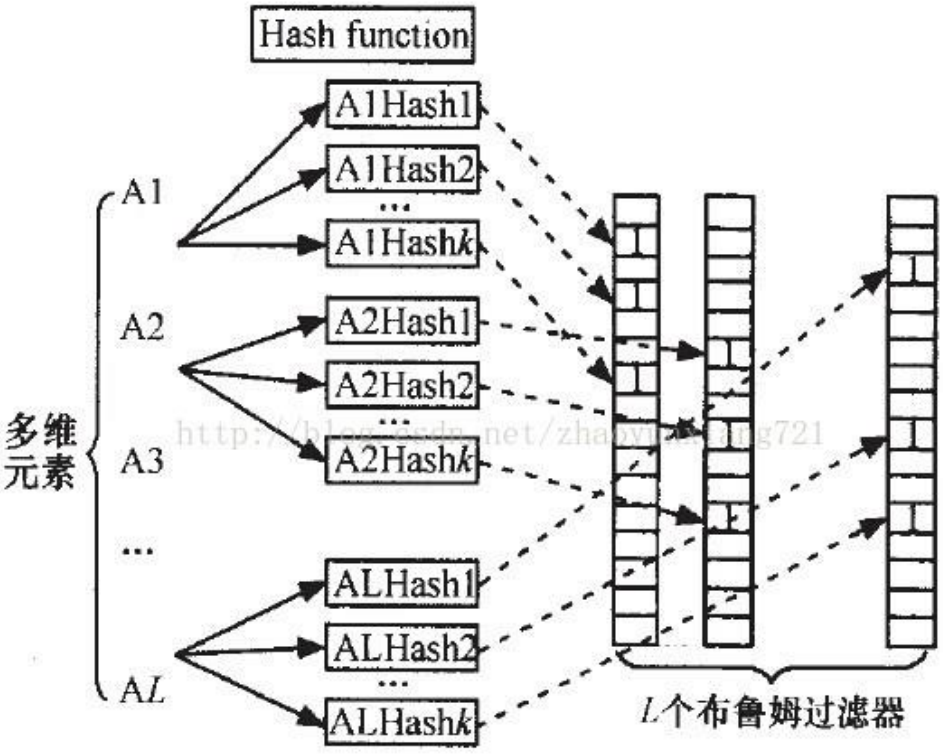
②：插入状态，为了将包含 n 个元素的集合 S={x1, x2,…,xn}和 Bloom filter 建立关系，Bloom filter 使用 k 个相互独立的哈希函数（Hash Function）（如图所示，使用了 3 个函数），k 个哈希函数分别将集合中的每个元素映射到{1,…,m}的范围中（例如，对任意一个元素 x，第 i 个哈希函数映射的位置 Hashi(x)就会被置为 1）。

③：查询状态，通过上述步骤，我们已将集合 S={x1, x2,…,xn}和 Bloom filter 建立起映射关系，接下来就通过查询来判断某个元素是否属于集合 S。如上图③所

示，我们对待查询的元素 y 通过 k 次哈希函数找寻对应的 k 个 bit 位，如果 k 个位置都是 1，那么我们就认为 y 是集合中的元素，否则就认为 y 不是集合中的元素（上图③中结果所示 y1 不属于当前集合，y2 属于当前集合）。

四、课设内容

针对多维元素的表示和查询问题，目前存在一种多维 Bloom Filter 解决方案。该方案采用和元素维数相同的多个标准 Bloom Filter 组成，直接将多维元素的表示和查询分解为单属性值子集合的表示查询，元素的维数有多少，就采用多少个标准的 Bloom Filter 分别表示各自对应的属性。进行元素查询时，通过判断多维元素的各个属性值是否都在相应的标准 Bloom Filter 中来判断元素是否属于集合。

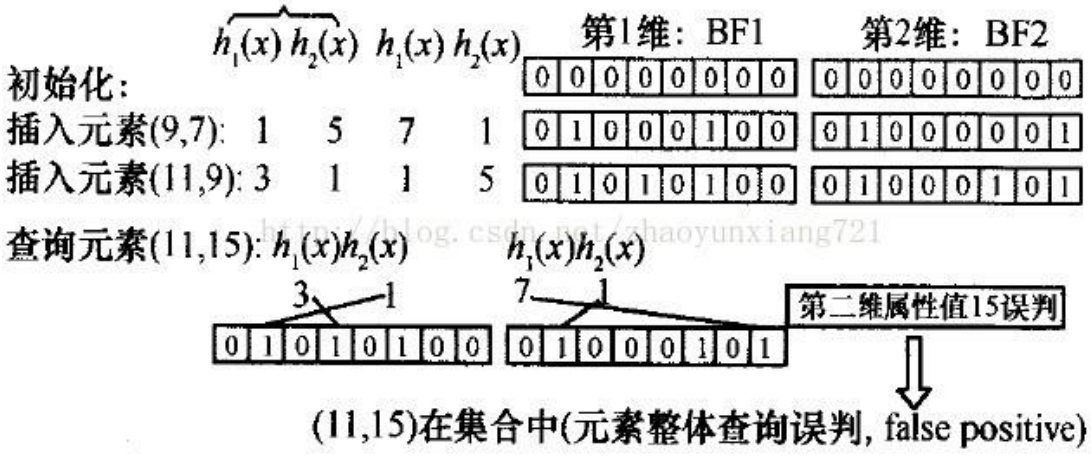


五、实验过程

对于{n，m，k，L}的 MDBF，判断元素是否从属集合，需要判断所有的属性值是否在对应的属性子集合，多维 Bloom filter 误判率为：

$$f^{MDBF}(m,k,n,l)=\prod_{i=1}^{L} f^{BF}(m,k,n)=(f^{BF}(m,k,n))^{L}$$

多维 Bloom filter 查询时间为 D，所需空间为 D(m×L)。下面给出多维的例子：设二维元素为{A1，A2)，其中 A1 和 A2 是 2 个不同的属性。使用 MDBF 需要 2 个标准 Bloom filterBF1、BF2 用来分别表示属性 A1 和属性 A2。单属性域的 Bloom filter 向量都取=8 bit。每次映射和查找的散列函数的个数为 2 个，2 个属性值映射的散列函数取一致，简单定义这 2 个散列函数为：h1(x)=x mod 8 和 h2(x)=(2x+3) mod 8。其中数据集合是{(9，7)，(11，9)}，需要查询的元素是(11，15)。表示集合之前，2 个向量都需要初始化。元素(9,7)插入后，第一维 Bloom filter 向量 BF1[1]和 BF1[5]置位，第二维 Bloom filter 向量 BF2[7]和 BF2[1]置位，两向量状态分别如下图第 2 行所示。那么元素(11，9)插入后的向量状态如下图第 3 行所示。



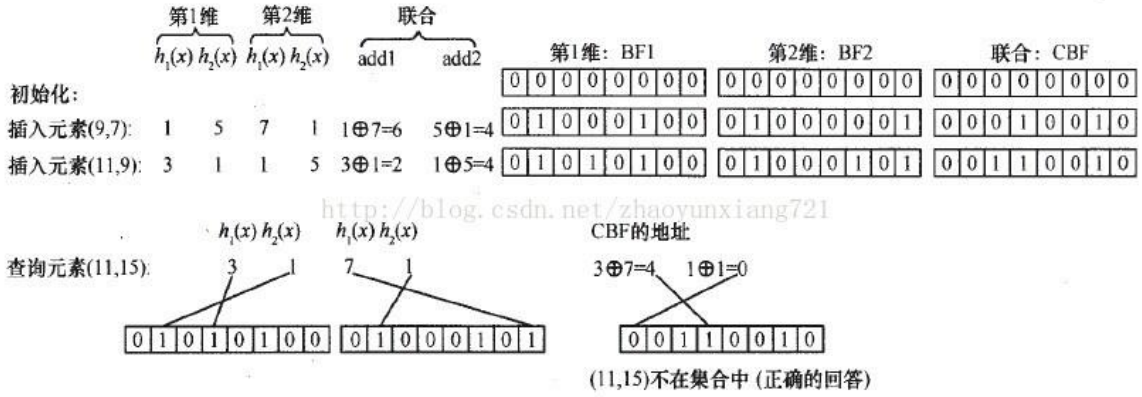
查询(11，15)是否在集合中。首先检查 11 是否在第一个属性集合中，11 对应的 2 个散列地址 3、1，发现 BF1[3]，BF1[1]都已置位，说明 11 在属性 A 集合中。然后检查 15 的 2 个散列地址，发现 BF2[7]，BF2[1]也已经置位，说明 15 也在属性 A2 的集合中，得出(11，15)在集合中的错误结论。“MDBF 出现误判的情况是每维都出现误判断时，元素才会被误判断”。通过上面的实例分析发现：实际上，MDBF 算法在任何一维误判断时都会导致元素的误判断。如例 2 所示，两维元素(11，15)进行查询判断时，第二维 15 却不在第二维子集中，但是使用 BF2 误判断 15 在该子集中，此时就出现了两维元素由于一维的误判而导致元素整体的误判。成为

MDBF 的缺陷。

上述结果的产生，是因为 MDBF 使用每个单独的 BF 来表示元素的每个单独的属性值，没有相应的结构将各属性值合成的元素表达出来。MDBF 分割了各个属性值属于元素一体的特点，仅通过单独判断元素的各个属性值是否在对应的子集合来进行元素的从属判断，不可避免发生例 2 的情况，将不属于集合的元素(11，15)误判为属于集合，因此有必要对 MDBF 进行改进。

联合多维 Bloom Filter

考虑到如果能将各个属性合为一体的特性在 Bloom filter 中表示出来，得到元素的整体信息必然可以减少由于单维属性误判而导致的元素整体误判的可能性。由此提出改进的联合多维 Bloom filter，它由两部分过滤器组成，第一部分是用标准 Bloom filter 表示的各属性子集，采用和 MDBF 一样的机制；第二部分是一个联合 Bloom filter,用来表示各个属性值的联合。这里存在如何将元素表示到 CBF 的问题，如果 MDBF 每次映射散列地址范围一致，那么可以直接用不同属性域的对应散列映射地址进行异或运算，通过 2 次散列运算获得元素在 CBF 中的位置，将 CBF 对应位置置位，完成元素到 cBF 的表示。下面用一个例子来解释算法思路，如下图 5 所示，其中，圆圈中加一个加号为异或运算。



重复上次的查询(11，15)是否是集合的元素。在多维 bloom Filte 的基础上，还需要进行元素整体是否在 CBF 的检查，元素(11，15)在 CBF 中的 2 个映射地址分为 addr1=3 异或 7=4，addr2=1 异或 1=0。虽然 CBF[4]=1，但是 CBF[0]=0，得出元素(11，15)不在集合中的正确结论。虽然 CMDBF 算法只在 MDBF 算法上增加了一个用于表示各属性联合的 CBF 过滤器，但元素在 CBF 的映射位置由所有的属性值共同决定，有效解决由于单属性的误判断而造成元素整体误判断的可能。

在误判率上，联合多维明显比之前的要好，不再证明。

六、课设总结

本次课设收获很大， 不仅加深了对课堂中学习的 Bloom filter 的理解，更学会了对 Bloom filter 的应用。多维数据的属性表示与索引在一开始让我很疑惑，不知如何下手，在查阅各种资料后，开始明白一点。最终理解了如何设计一个基于 Bloom filter 的多维数据属性表示与索引

本次课设的遗憾是：由于时间关系，没能理解更多大数据存储相关的知识原理。

这门课最开始吸引我的是华老师在讲台上讲 flash 存储的特性，以及相变存储之类是如何实现的。这些用神奇的材料特性来实现 0、1 两种状态进而实现存储的过程使我非常惊讶和着迷，尽管硬件研究在目前看来仍然是枯燥单调的。希望大数据存储系统与管理这门课越办越好！感谢华老师和施老师的细心授课！

参考文献

F. Bonomi, M. Mitzenmacher, R. Panigrahy, S. Singh, and G. Varghese, “Beyond Bloo m Filters: From Approximate Membership Checks to Approximate State Machines,” Pro c. ACM SIGCOMM, 2006. • Y. Zhu and H. Jiang, “False Rate Analysis of Bloom Fil ter Replicas in Distributed Systems,” Proc. Int’l Conf. Parallel Processing (ICPP ’06), p p. 255-262, 2006. • S. Dharmapurikar, P. Krishnamurthy, and D.E. Taylor, “Longest Pre fix Matching Using Bloom Filters,” Proc. ACM SIGCOMM, pp. 201-212, 2003. • L. F an, P. Cao, J. Almeida, and A. Broder, “Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol,” IEEE/ACM Trans. Networking, vol. 8, no. 3, pp. 281-293, Ju ne 2000. • B. Xiao and Y. Hua, “Using Parallel Bloom Filters for Multi- Attribute Re presentation on Network Services,” IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 1, pp. 20-32, Jan. 2010. • Y. Hua, Y. Zhu, H. Jiang, D. Feng, and L. Tian, “Scalable and Adaptive Metadata Management in Ultra Large-scale File Systems,” Proc. 28th Int’l Conf. Distributed Computing Systems (ICDCS ’08), pp. 403-410, 2008. • D. Guo, J. Wu, H. Chen, and X. Luo, “Theory and Network Application of Dynamic Bl oom Filters,” Proc. IEEE INFOCOM, 2006.