相似性—局部性方法相关参数分析

张星煜,张 建,辛明军
(上海大学 计算机工程与科学学院,上海 200444)

摘 要:大数据时代到来,备份数据量增大给存储空间带来新的挑战。重复数据删除技术在备份存储系统中正逐渐流行,但大量数据访问,造成了磁盘的很大负担。针对重复数据删除技术存在的块索引查询磁盘瓶颈问题,文中提出了文件相似性与数据流局部性结合方法改善磁盘 I/O 性能。该方法充分发挥了各自的优势,相似性优化了索引查找,可以检测到相同数据检测技术不能识别的重复数据;而数据局部性保留了数据流的序列,使得 cache 的命中率提高,减少磁盘访问次数。布鲁过滤器存储数据块索引可节省大量查询时间和空间开销。对于提出的解决方法所涉及的重要参数如块大小、段大小以及对误判率的影响做了深入分析。通过相关实验评估与性能分析,实验数据与结果为进一步系统性能优化问题提供了重要的数据依据。

关键词:重复数据删除技术;相似性与局部性;布鲁过滤器;磁盘瓶颈
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2014)11-0047-04
doi:10.3969/j.issn.1673-629X.2014.11.012

Analysis of Related Parameters Based on Similarity-locality Approach

ZHANG Xing-yu, ZHANG Jian, XIN Ming-jun
(School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China)

Abstract: Big data era comes, and the increase of the backup brings new challenges to deduplication. Data deduplication is becoming increasingly popular in storage systems to data backup, but a lot of accesses cause a great burden of disk. For the block index-lookup disk bottleneck, present that combining file similarity with data stream locality is to improve disk I/O performance, and the approach reaches their full advantages. Similarity optimizes index-lookup and detect the duplicate data cannot be recognized by duplicate data detection technology. Locality reserves the sequence of the data stream, and it improves the hit rate of cache and reduces disk access. Bloom filter stores block index to save a lot of time and space overhead. The related parameters of the solution are made deep analysis, such as the block size, the segment size, and their sizes influence to false positive. Through the relevant experiment assessment and performance analysis, the experimental data and results provide an important basis for the further system performance optimization problem.

Key words: data deduplication technique; similarity-locality; Bloom filter; disk bottleneck

0 引 言

信息化时代,数据呈现爆炸性增长,而备份和归档系统中大量数据冗余,给存储带来了巨大挑战[1]。重复数据删除技术可以降低存储空间巨大开销。为了充分利用重复数据删除技术,最大限度消除冗余,提高系统吞吐率,研究者们提出了一系列的解决策略。如减少磁盘访问次数,摘要向量技术(bloom filter),基于局部性缓存机制,基于文件相似性策略,SSD(Solid State Disk,固态硬盘)策略等。文中采用的是数据流局部性与文件相似性结合的方法,并采用 bloom filter 存储索引,以此减少存储空间和磁盘访问次数,降低系统性能开销。

对于 bloom filter 相关参数进行了设置与分析研究,并对数据块相似性与局部性结合方法相关重要参数进行分析与设置,如数据段大小、Block 大小都对系统性能有很大影响。通过模拟实验,分析这些参数影响因素以及如何设置参数大小,为更进一步的索引优化工作提供了重要的数据基础。

1 重复数据删除技术

1.1 概 述

重复数据删除技术是一种数据缩减技术,通常用

收稿日期:2013-12-22 **修回日期**:2014-03-25 **网络出版时间**:2014-09-11
基金项目:国家自然科学基金资助项目(61074135)
作者简介:张星煜(1988-),女,硕士研究生,研究方向为大数据去除冗余。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140911.1001.027.html

于存储备份和归档系统中。当相同数据段的多个副本需要存储时,重复数据删除技术仅存储数据段的一个副本,过滤掉相同的数据段,仅仅使用一个指针指向已存储的数据段。

重复数据删除技术主要分为两大类,相同数据检测技术和相似数据检测技术[2]。

相同数据检测技术以不同粒度可以分为文件级和数据块级,通过查找比对文件和数据块检测相同数据。文件级相同数据检测技术是完全文件检测(Whole File Detection,WFD)技术,这种技术主要通过 hash 技术查找比对文件并检测出相同文件。对于数据块相同数据检测技术,根据相同数据块的细粒度划分为固定分块(Fixed-Sized Partition,FSP)检测技术、可变分块(Content-Defined Chunking,CDC)检测技术、滑动块(Sliding Block)技术进行重复数据的查找与删除。

相似数据检测技术主要通过 shingle 技术、bloom filter 技术和模式匹配技术挖掘出相同数据检测技术不能识别的重复数据;采用 delta 技术对相似数据进行编码,最小化压缩相似数据,缩减存储空间和网络带宽占用。

1.2 重复数据删除系统磁盘瓶颈问题

为了节省成本,会使用少量的内存,因而数据索引不可能全部存放在内存,多数数据索引都存储在磁盘,这将导致大量的磁盘访问[3],降低了存储系统数据访问的性能。为了获得高吞吐率、低开销的相同块删除存储系统,出现了一些减轻磁盘瓶颈的技术,如摘要向量技术、基于流的块排列技术和局部性保持技术[4]等。文中针对磁盘瓶颈问题提出相似性-局部性方法,以解决磁盘读写性能,提高系统性能,而相关参数的设置对于系统性能尤为重要。

数据块和数据段大小是两个需要确定的关键参数,两个参数的设置直接影响到磁盘索引空间、数据块查询磁盘瓶颈问题和重复数据删除率[5],因此需要实验分析进行参数设置。重复数据删除系统消重能力可以用重复数据删除率来衡量,而重复数据删除率依赖于数据集自身的特征、数据划分策略和数据块大小[6]。

2 布鲁过滤器

布鲁过滤器是由 Howard Bloom 于 1970 年提出的二进制向量数据结构[7],它是一种高效简洁,可以表示数据集合,并支持集合查询的数据结构。布鲁过滤器是一个长度为 m 的位向量和一个由 k 个 hash 函数构成的 hash 函数组。各 hash 函数相互独立且函数的取值范围为 $\{0,1,\cdots,m-1\}$,n 个元素的数据集合 $S=\{s_1,s_2,\cdots,s_n\}$,通过 k 个 hash 函数 $h_1,h_2,\cdots,h_k$ 映射到位向量 $BF=(b_1,b_2,\cdots,b_m)$ 中。位向量 BF 就是集合 S 的布鲁过滤器表示,记为 $BF(S)_{m,k}$,或简记为 $BF(S)$。

初始化 BF,每位值都为 0。当插入元素 x 到集合 S 时,计算 x 对应的 k 个 hash 地址 $h_1(x),h_2(x),\cdots,h_k(x)$,并将 hash 值对应的 BF 位置为 1($bh_j(x)=1,1\leqslant j\leqslant k$)。查询给定的元素 x 是否属于集合 S,计算 x 对应的 k 个 hash 地址 $h_1(x),h_2(x),\cdots,h_k(x)$,然后检查向量 BF 对应的 k 个位置是否全为 1,若全为 1,则认为 x 属于集合 S;若 k 个位置任一位为 0,则 x 必定不属于集合 S。如图 1 所示,$m=18$,$k=3$,集合 $\{x,y,z\}$,元素 w 不属于集合 $\{x,y,z\}$。

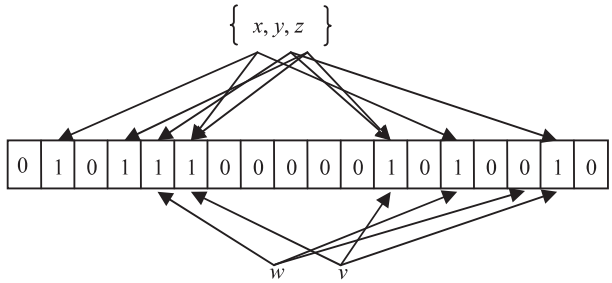

图 1 示例图

但是可能将不属于集合的元素判为集合元素,如图中的 v 元素。这种情况被称作假阳性(false positives),其假阳性概率为

$$p=(1-e^{-(m/n\ln 2)\cdot n/m})^{(m/n\ln 2)}$$

由公式可以看出,假阳性概率可由向量长度 m、集合元素规模 n 和 hash 函数数目变化来控制。

当向量长度 m 和集合元素规模 n 一定时,布鲁过滤器的假阳性概率最小时最优的 hash 函数数目 k 值如下:

$$k=\frac{m}{n}\ln 2$$

布鲁过滤器存储空间和插入/查询时间都是常数。布鲁过滤器存储空间长度 m 位,可存储 n 个元素($m\ll n$),大大节省存储空间。插入元素只计算 k 个 hash 函数,进行 k 位的查找,其时间复杂度均是常数 $O(k)$。

用 Bloom filter 数据结构来存储指纹可以减少查找不在磁盘中的块的次数。Bloom filter 存储在内存中,存储数据块指纹。如果在 Bloom filter 中查找不到一个块的索引,说明数据块不存在,可直接存储数据块相关信息和元数据本身,则不需要进一步查找,减少了磁盘 I/O 访问次数。如果 Bloom filter 查找到块索引,那么很有可能这个块就在块索引内,但并不是一定的(可能存在误判)。由于 Bloom filter 是存储在内存中的数据结构,当系统关机后,会将其写到 disk 上,启动后,又会从 disk 上读入内存中。为了处理停电和不正常的关机情况,系统会周期性地将 Bloom filter 备份到磁盘上,并设置检查点。当恢复时,系统会载入最近备份的副本,并处理从最近一个检查点添加到系统中的数

据,将其信息加入到 Bloom filter 中。

3　相似性–局部性算法

Bhagwat 等人提出利用文件相似性能优化索引查询,并设计了 Extreme Binning 策略[8]。文件相似性根据 Broder 等的最小值独立置换理论[9],只需要比对文件内最小的数据块指纹值就可以判断两个文件是否相似,通过建立文件索引和以文件粒度分组的块索引构成的二级索引结构,使得每个文件的数据块查询只需要一次磁盘访问。数据流存在重复局部性[10],当一个数据块在数据流中重复出现时,该数据块在旧数据流中邻近的其他数据块也很可能在新数据流中重现。利用这种局部性,保留数据流原有的顺序,可以减少磁盘访问次数,查找出更多重复数据[11]。

当备份数据流进入,根据文件大小会被分割成许多段。一个大文件被分割成多个小的段。许多小的文件组成一个段。而一个数据块由多个连续的数据段组成,可以保留数据段的局部性。并在每个数据块中设置指纹表存储连续的数据段指纹,为每个数据块选择一个数据段指纹存储到内存指纹表中。相似性—局部性算法思想就是一个数据块指纹首先与内存指纹表进行相似性比较,若没有发现相似性大的数据段指纹,则认为是新的数据段指纹,进行存储;当发现最相似的数据段指纹时,查看数据段相应的数据块是否在内存中,若在内存,则与数据块内的所有数据段指纹再比较是否相同;若数据块不在内存,则根据数据块位置把相应的数据块从磁盘调到内存中。

Bloom filter 存储指纹,因此两个段 Bloom filter 可以比较相似性。当两个段有越多的共同位为 1,它们就越相似。这里使用 Tanimoto 相似性测量来论证文中的方法,这个测量方法对应的是共同位设置为 1 的数量与所有被设置为 1 的数量之比。给出两个指纹 Bloom filter 向量 $\boldsymbol{A}$ 和 $\boldsymbol{B}$。

$$\mathrm{Sim}_{\mathrm{T}}(\boldsymbol{A},\boldsymbol{B}) = \frac{\text{Both } \boldsymbol{AB}}{\text{Only } \boldsymbol{A} + \text{Only } \boldsymbol{B} + \text{Both } \boldsymbol{AB}}$$

其中,Only $\boldsymbol{A}$ 表示 Bloom filter 向量 $\boldsymbol{A}$ 设置为 1 的位数量;Only $\boldsymbol{B}$ 表示 Bloom filter 向量 $\boldsymbol{B}$ 设置为 1 的位数量;Both $\boldsymbol{AB}$ 表示向量 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的 Bloom filter 都设置为 1 的位数量。相似性值范围为[0.0,1.0],这个值越大越相似。当 $\mathrm{Sim}_{\mathrm{T}}(\boldsymbol{A},\boldsymbol{B})$ 的值为 1 时,则说明 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 相同。设置只有相似性值大于 0.8 时,才会进一步查找是否有相同数据段。

采用相似性–局部性方法减少了磁盘访问次数[12],由于采用数据流局部性原理,将序列连接的数据段存储到同一个数据块中。当在内存通过比较数据段 Bloom filter 相似性时,相似性大,则从磁盘调取相应的数据块到内存,只需要一次磁盘访问,可调取相邻数据段,磁盘访问次数明显减少,系统可进一步比较数据块中的数据段相似性。Bloom filter 相似性比较速度快,提高了系统性能。

4　实验评估

对于提出的解决方案,需要对一些性能参数进行分析研究,为下一步性能研究与优化提供重要依据。首先是块与段的大小参数的确定,提高重复数据删除率;其次是这样的参数大小对布鲁过滤器假阳性概率的影响[13],从而选出最优的参数,使得假阳性概率可以接受,而重复数据删除率高。此次实验收集了两组数据集,第一组数据集验证块和段的大小对重复数据删除率的影响,从而选出最优的 2 个。然后对于最优的两个参数值在第二组数据集上进行实验,查看对于假阳性概率的影响,从而选出最优的数据块和段[14]。

4.1　第一组数据集实验结果与分析

图 2 显示了第一组数据集针对不同数据块和数据段大小对重复数据删除率的影响[15]。由于重复数据删除率高,不能看出明显的变化趋势。图中 y 轴表示重复数据未删除率,由图可明显看出随着块的增大,重复数据删除率明显提高,这是因为数据块越大,越能好地保留数据集的局部性,从而发现更多重复数据段。当数据块大小一定时,段越小重复数据删除率越高,原因在于数据集划分的越小,可发现更多的相同数据段。

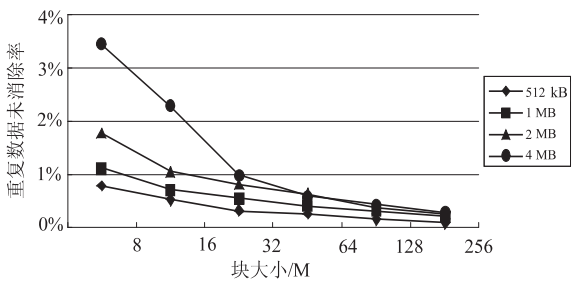

图 2　块和段大小对重复数据删除率的影响

图 3 是数据块和数据段大小变化对消除重复数据时间花费的影响。由图中可以看到,当数据块大小一定时,数据段越小,消除重复数据时间花费呈递增趋势。原因在于数据段越小,块存放的数据段越多,相似性比对的次数增加,花费时间就越多。当数据段一定,数据块增大时,消重时间花费递增。虽然数据段越小重复数据删除率越高,但同一数据集划分的段数也越多,在给定的布鲁过滤器出现假阳性概率也会提高。选出最优的两个数据段是 1 M 和 2 M,因为重复数据删除率高同时消重时间花费适中。块大小选定为 256 M,由图可以看出重复数据删除率很高,减少了磁盘访问次数,系统读写性能也相应提高,有利于删除更多的重复数据。

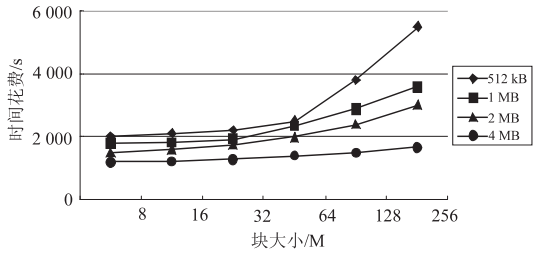

图 3 块和段大小变化时消重时间花费情况

4.2 第二组数据集实验结果与分析

通过第一组数据集实验结果,对第二组数据集测试不同数据段大小对于布鲁过滤器误判率的影响,从而选出合适的数据段大小参数[16],为后续的其他系统性能研究提供实验依据。第二组数据集采用递增数据集。如图 4 所示,随着数据集数量的增大,误判率增大,由于数据集越大,划分的数据段也越多,当布鲁过滤器设置参数一定时,存入越多数据段引起误判率的条件概率增大。由实验结果显示,1 M 数据段比 2 M 数据段的误判率高。

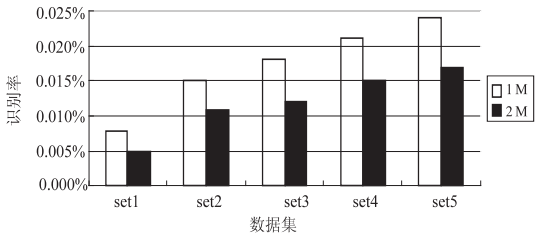

图 4 数据集与数据段变化对误判率的影响

由两组实验结果,得出数据段为 2 M 和数据块为 256 M 时,数据集的重复数据删除率高,相应地消重时间花费适中,而布鲁过滤器的误判率降低。通过实验,得出可靠的相关参数数据,为系统的性能分析提供了重要依据。

5 结束语

重复数据删除技术的应用很广泛,对于不同的应用及出现的挑战提出了不同的方法。文中针对重复数据删除技术对于磁盘瓶颈问题提出了相关的方法理论。对于提出的方法,需要一些重要参数,而这些参数值的选择会对系统性能产生不同的影响。针对提出的方法,实验评估了相关参数,通过实验结果给出分析,选择合适的参数值,为后续研究性能提供依据。

对于提出的方法将应用到提出的重复数据删除系统框架模型中,针对不同的性能,消重率、吞吐率、内存占用情况等需求,选择不同的参数来提高系统重复数据删除率。针对磁盘瓶颈问题,随着科技的发展,存储设备也在更新,出现了很多新型的存储介质,如闪存(Flash)、相变存储器(Phase Change Memory,PCM)、固态硬盘(Solid State Disk,SSD),这些存储介质有很多优势,可以解决磁盘瓶颈。在减轻磁盘瓶颈的同时,对于判断重复数据,数据搜索速度,存储空间利用率等的影响,需要综合考虑,使之达到一个很好的平衡点。

参考文献:

[1] Manyika J,Chui M,Brown B,et al. Big data:the next frontier for innovation,competition,and productivity[M]. [s. l.]:McKinsey Global Institute,2011.

[2] 敖 莉,舒继武,李明强. 重复数据删除技术[J]. 软件学报,2010,21(5):916-929.

[3] Zhu B,Li K,Patterson H. Avoiding the disk bottleneck in the data domain deduplication file system[C]//Proc of USENIX FAST. [s. l.]:[s. n.],2008.

[4] Lillibridge M,Eshghi K,Bhagwat D,et al. Sparse indexing:large scale,inline deduplication using sampling and locality[C]//Proc of the 7th USENIX conference on file and storage technologies. San Francisco:USENIX,2009:111-123.

[5] 付印金,肖 侬,刘 芳. 重复数据删除关键技术研究进展[J]. 计算机研究与发展,2012,49(1):12-20.

[6] 陆游游,敖 莉,舒继武. 一种基于重复数据删除的备份系统[J]. 计算机研究与发展,2012,49(S):206-210.

[7] Bloom B H. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM,1970,13(7):422-426.

[8] Bhagwat D,Eshghi K,Long D,et al. Extreme binning:scalable,parallel deduplication for chunk-based file backup[C]//Proc of MASCOTS. [s. l.]:[s. n.],2009.

[9] Broder A,Charikar M,Frieze A,et al. Min-wise independent permutations[J]. Journal of Computer and System Sciences,2000,60(3):630-659.

[10] Koller R,Rangaswami R. I/O deduplication:utilizing content similarity to improve I/O performance[J]. ACM Transactions on Storage,2010,6(3):211-224.

[11] Gupta A,Pisolkar R,Urgaonkar B,et al. Leveraging value locality in optimizing NAND flash-based SSDs[C]//Proceedings of the 9th conference on file and storage technologies. [s. l.]:USENIX Association,2011.

[12] Xia Wen,Jiang Hong,Feng Dan,et al. Silo:a similarity locality based near-exact deduplication scheme with low RAM overhead and high throughput[C]//Proc of USENIX ATC. [s. l.]:USENIX,2011.

[13] 曾 涛. 重复数据删除技术的研究与实现[D]. 武汉:华中科技大学,2011.

[14] 贾志凯,王树鹏,陈光达,等. 一种并行层次化的重复数据删除技术[J]. 计算机研究与发展,2011,48(S):100-104.

[15] 李 超,周晓阳,王树鹏,等. 基于二级索引的重复数据删除系统中性能相关参数的量化分析与研究[J]. 计算机研究与发展,2012,49(S):173-177.

[16] 李 超,王树鹏,云晓春,等. 一种基于流水线的重复数据删除系统读性能优化方法[J]. 计算机研究与发展,2013,50(1):90-100.

相似性-局部性方法相关参数分析

作者： 张星煜， 张建， 辛明军， ZHANG Xing-yu， ZHANG Jian， XIN Ming-jun
作者单位： 上海大学 计算机工程与科学学院,上海,200444
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(11)