

冗余数据删除技术

莫柠源¹⁾

¹⁾(华中科技大学武汉光电国家研究中心 武汉 430074)

摘要 在全球数据量急剧增长的背景下，存储系统中的冗余数据问题成为了一个不容忽视的挑战。研究表明，用户桌面文件系统中的重复数据量大约占 42%，这突显了有效重复数据删除技术的必要性。为了解决这一问题，数据去重、差量压缩和无损压缩技术被提出并广泛应用。数据去重技术，通过对文件或数据块进行哈希签名来识别冗余，是一种粗粒度的数据减少方法，能够显著降低企业存储系统的数据占用。差量压缩技术则在更细的粒度上工作，通过存储文件的变化部分来进一步减少信息冗余。这些技术虽然有效，但在特定场景下仍有改进的空间。在此背景下，DeepSketch、Light-Dedup、LoopDelta 和 PFC-delta 等先进技术的出现，提供了解决存储系统中冗余数据问题的新途径。DeepSketch 采用机器学习方法，通过深度神经网络创建数据块的近似数据签名，大大提高了后去重差量压缩的参考搜索精度。Light-Dedup 则为非易失性内存（NVM）文件系统提供了一种轻量级的内联去重框架。通过特殊的 Light-Redundant-Block-Identifier（LRBI）和一个精心设计的元数据表（LMT），Light-Dedup 能够显著降低元数据 I/O 放大，并提升去重性能。LoopDelta 利用双重局部性相似度跟踪、局部性感知预取、缓存感知过滤器和逆向增量压缩来提高压缩比和恢复性能，同时保持备份吞吐量。PFC-delta 针对打包数据集的去重和增量压缩，提出了的方法减少了额外的 I/O 操作和计算开销，同时提升了压缩比和数据恢复速度。这些创新技术与传统的去重和压缩方法相结合，不仅克服了各自的局限性，还发挥了它们之间的互补性。这种综合的方法不但能提高存储系统中的数据处理能力，而且还能在不牺牲系统性能的前提下，实现更彻底的重复数据清除。这对于降低存储成本和提升数据中心的效率至关重要，使其能够有效应对未来数据量的持续激增。

关键词 差量压缩；相似性检测；数据去重；特征提取

Data Deduplication Technology

Ningyuan Mo¹⁾

¹⁾(Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, 430074, China)

Abstract In the context of a rapidly increasing global data volume, the issue of redundant data in storage systems has emerged as a challenge that cannot be overlooked. Studies have indicated that approximately 42% of the data in user desktop file systems is duplicative, highlighting the necessity for effective data deduplication technologies. To address this issue, techniques such as data deduplication, delta compression, and lossless compression have been proposed and widely applied. Data deduplication technology, which identifies redundancy through hashing signatures of files or data blocks, is recognized as a coarse-grained method of data reduction, capable of significantly reducing the data footprint in enterprise storage systems. Delta compression technology operates at a finer granularity, further reducing information redundancy by storing only the changes in files. Although these technologies are effective, they still have room for improvement in specific scenarios. Against this backdrop, the emergence of advanced technologies such as DeepSketch, Light-Dedup, LoopDelta, and PFC-delta offers new avenues for addressing the issue of redundant data in storage systems. DeepSketch employs a machine learning approach, creating approximate data block signatures using deep neural networks, thus greatly enhancing the precision of reference searching in post-deduplication delta compression. Light-Dedup provides a lightweight inline deduplication framework for non-volatile memory (NVM) file systems, utilizing a specialized Light-Redundant-Block-Identifier (LRBI) and a meticulously designed metadata table (LMT) to significantly reduce metadata I/O amplification and enhance deduplication performance. LoopDelta is characterized by the utilization of dual locality

similarity tracking, locality-aware prefetching, cache-aware filtering, and reverse incremental compression to improve compression ratios and recovery performance while maintaining backup throughput. PFC-delta, targeting deduplication and delta compression for packed data sets, proposes a method that reduces additional I/O operations and computational overhead while enhancing compression ratio and data recovery speed. These innovative technologies, combined with traditional deduplication and compression methods, not only overcome their respective limitations but also exploit their complementarity. This integrated approach not only enhances the data processing capabilities in storage systems but also achieves more thorough deduplication without sacrificing system performance. This is vital for reducing storage costs and enhancing the efficiency of data centers, enabling them to effectively cope with the continuous surge in data volumes in the future.

Key words Delta Compression; Resemblance Detection; Data Deduplication; Feature Extraction;

1 绪论

1.1 研究背景和意义

随着信息技术的蓬勃发展和互联网的不断普及,企业及个人每天都会产生大量的数据。根据 IDC 的统计和预测,2018 年全球数据产生量达到了 33 ZB,而到了 2025 年,这一数字将达到惊人的 175 ZB,全球数据总量将迎来更大规模的爆发^[1]。

为了解决存储占用过多,人们提出了冗余数据消除技术。一般来说,无损的冗余数据消除技术可以分为三种:数据去重、增量压缩和传统压缩。传统压缩技术只能在有限的压缩窗口内查找冗余数据;增量压缩则可以通过相似性检索技术在整个存储系统内找相似的文件(或数据块),然后在相似文件(或数据块)之间查找冗余数据;数据去重则直接通过索引技术在整个存储系统内进行重复数据块的消除。

传统压缩技术的压缩粒度是字符和字符串。早期的压缩技术主要依赖对数据进行编码来减少存储空间,代表技术是哈夫曼编码^[2]。这种编码技术压缩效果有限,后来出现更复杂的压缩算法与技术来提高压缩率,比较经典的有 LZ 系列的压缩算法^{[3][4]}。然而由于其只能在有限的压缩窗口内查找重复数据,本文不作过多的考虑;

数据去重是一种无损压缩技术,通过维护重复数据的一个副本,消除重复的写数据以提高空间利用率,因此数据去重也被称作为单例存储。不同于以字符或字符串为单位的传统压缩技术,数据去重技术以文件或数据块为单位,实现了冗余数据的快速删除,很好地满足了企业对于冗余数据删除的吞吐量的需求。为了高效地消除存储设备中的冗余数据,降低存储开销,越来越多的企业已经将数据去重技术运用到存储系统中。然而,根据 EMC 研究

数据,经过数据去重后的存储系统中依旧存在大量的重复数据。这是因为数据去重只能实现粗粒度的冗余数据删除,无法消除数据块内部细粒度的冗余数据。

增量压缩技术是一种细粒度的无损压缩技术,可以通过编码的方式有效地消除相似文件或相似数据块内部的数据冗余。由于两种技术具有互补性,增量压缩可以在数据去重的基础上进一步消除冗余数据,实现更好的冗余数据删除效果。

因此,本文主要介绍数据去重和增量压缩技术。

1.2 相关技术研究概述

增量压缩技术是一种细粒度的无损压缩技术,压缩对象是两个相似的文件或数据块。具体来说,增量压缩会利用滑动窗口技术寻找两个相似对象之间的重复数据。对于重复的数据内容增量压缩会使用 COPY 命令进行编码,仅记录重复部分的位置和长度信息。对于不重复的数据内容,增量压缩会使用 INSERT 命令进行编码,记录不重复部分的数据内容和长度。以数据块 A 和 B 为例,假设数据块 A 是已经存储过的数据块,而数据块 B 是数据块 A 的相似数据块(通过 sketch 比对),那么增量压缩可以根据数据块 A 的内容对数据块 B 进行增量编码,从而只保存数据块 B 中与数据块 A 不重复的数据内容。在增量编码的过程中,被压缩的数据块被称为目标块,而另一个完全存储的数据块被称为引用块。

相似性检测是增量压缩的关键组成部分。主要使用文件或数据块的特征值来代替数据内容进行相似度判断。这一类相似性检测方法一般分为特征提取和特征匹配两个步骤。特征提取指的是从文件或数据块中提取出具有代表性的特征值的过程,而特征匹配指的是通过匹配特征值的方式查找相似文件或数据块的过程。目前存在不同的相似性检

测算法, 例如 Super-feature^[5]、Finesse。

数据去重技术使用安全哈希摘要算法来识别并消除存储设备中的冗余数据。不同于以字符或字符串为单位的传统压缩技术, 数据去重技术以文件或数据块为单位, 实现了冗余数据的快速删除, 很好地满足了企业对于冗余数据删除的吞吐量的需求。

目前结合了相似性检测和数据去重技术的增量压缩方案主要有:

SIDC^[6]利用局部性进行重删和压缩。构建一个类似于重复数据删除系统的缓存和索引系统 (即 Bloom 过滤器和索引), 在将前一个流的指纹加载到缓存中时, 也从同一流加载草图。也就是数据块的超级特征值和指纹共同保存在容器的元数据区中。。

DARE^[7], 一种低开销的重复数据删除感知相似性检测和消除方案, 它有效地利用现有的重复邻接信息, 在基于重复数据删除的备份/归档存储系统中进行高效的相似性检测。DARE 背后的主要思想是采用一种方案, 调用基于重复邻接的相似性检测 (DupAdj), 如果任意两个相邻的数据块在重复数据删除系统中是重复的, 则考虑任意两个数据块是相似的 (即增量压缩的候选数据块), 然后通过改进的超特征方法进一步提高相似性检测效率。

DAC^[8]提出了一种新的去数据删除辅助压缩方案 (DAC) 方案, 集成了现有的数据缩减技术, DAC 大大减少了发送到 ssd 的写通信量。重复数据删除只考虑块之间精确匹配的值, 而无损压缩消除了块内精确匹配/重复的位模式, 而 DAC 则考虑了整个存储空间中的数据相似性。通过以更细粒度的方式利用重复数据删除引擎的散列函数 DAC 可以定位最相似的页面, 而不是完全完全相同的页面, 以克服重复数据删除的限制。一旦选择类似的页面作为写请求的参考, DAC 就会在请求的页面和参考页面之间执行 XORing, 结果数据在被无损压缩引擎压缩后被写入 flash。不同于增量压缩只考虑一个原始页面作为参考, DAC 在整个存储中搜索最相似的页面, 从而实现了更好的数据缩减。与简单的重复数据删除和无损压缩集成相比, DAC 减少了高达 30% 的写入数据量。

DeepSketch^[9]提出了一种新的用于重复数据删除后增量压缩的参考搜索技术。现有的重复数据删除后增量压缩技术提供与最佳数据缩减率相比, 数据缩减率明显较低。主要是由于增量压缩的参考搜

索精度有限, 无法为许多传入的数据块识别出良好的参考块。DeepSketch 使用深度神经网络提取数据块的 sketch, 利用深度学习的方法来实现增量压缩, 获得更高的搜索精度, 从而提高数据缩减效率。

2 数据去重技术

传统的数据去重过程的四个阶段: 数据分块、指纹计算、指纹索引和存储管理。其中数据分块和哈希计算占用大量的 CPU 资源, 这成为数据去重系统中的潜在的瓶颈。

2.1 数据分块

块级的数据去重技术有定长分块和变长分块 (即基于内容分块) 两种, 这里定长分块算法是最简单的实现方式, 即根据文件内容的偏移位置决定分块; 而基于内容分块算法是基于文件的内容来分块, 这样有效地解决了文件修改导致的数据内容偏移的问题。相对而言基于内容分块的数据去重技术更能适应内容频繁修改的负载, 所以能够发现和查找到更多的冗余数据。但是基于内容分块算法的计算开销更大, 因此常被用在延迟不敏感的场景, 如备份和归档存储系统。

现有主流的数据去重技术采用了基于 Rabin 的分块算法、基于 SHA-1 的指纹算法, 这使得数据去重技术在压缩存储空间的同时, 不可避免地带来了计算开销和时延^[10]。对此, 一般有两种优化方式:

(1) 控制分块过程中的最大块和最小块。因为这种分块算法决定分块的条件是基于数据内容的 Rabin 哈希值的模数判断的, 所以通过使用两个或者多个模数有益于增大 Rabin 指纹满足分块条件的概率, 从而产生更加均匀和收敛的数据块大小分布。虽然这种方法极其简单易行, 但是其对数据压缩率改进的效果很有限。还可以直接设定分块最大值与最小值。

(2) 对去重后的数据块进一步分块去重。由于基于内容分块算法具有一定的随机性, 其不能准确地来判断数据修改和没有修改的位置, 所以研究人员提出了对去重后的数据进一步分块处理, 通过进一步分成更小的数据块来找到更多的重复数据。这种方法较第一种方法复杂, 而且带来了存储管理和数据恢复的难度, 但是获得的数据压缩改进效果很好。

常见的数据分块算法如下:

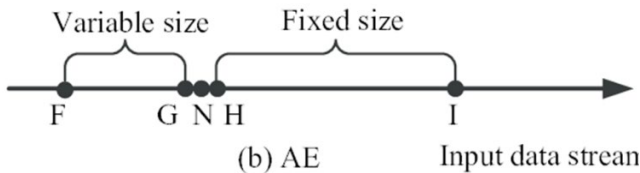


图 2-1 MAXP 算法

(1) MAXP：将文件或数据流看成由一串数字组成，并且尝试寻找局部的极大值作为块边界。极大值只依赖内容 而不依赖任何预定义参数，因此可以减少低熵字节串的影响^[11]。

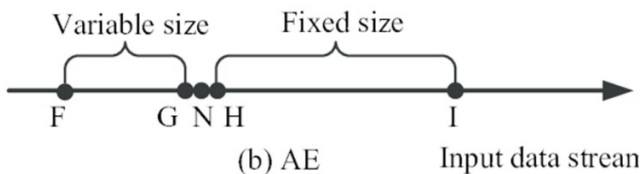


图 2-2 AE 算法

(2) AE：MAXP 需要回溯部分字节串才能准确地找到下一个极值点，因此提出了 MAXP 的变种 AE 分块算法。AE 不将极值点直接作为块边界，而是取一定偏移 位置的点作为块边界来避免回溯^[12]。

步骤1:输入数据流

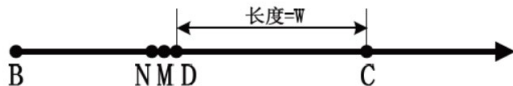
B是数据流第一个字节，从B开始处理



步骤2:找最大值点

M满足如下条件则是最大值点：

- 1) 区间 [B, N] 为空，或者 M 拥有区间 [B, M] 中的最大值；
- 2) M 的值不小于区间 [D, C] 中的所有值。



步骤3:输出块边界

将正在处理的字节C作为数据块边界输出。B'是剩下的数据流的第一个字符。

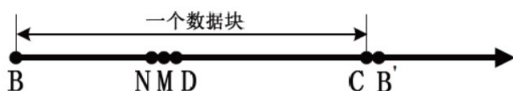


图 2-3 AE 算法步骤

2.2 指纹计算

传统的判断数据块是否相同的方法是对数据块内容进行逐字节的比较。这种方法过于耗时，不适合应用于大规模存储系统中。为了解决这个问

题，数据去重技术使用数据块的安全哈希摘要来代替数据块内容进行匹配，极大地简化了重复数据检测的过程。由于数据块的哈希摘要可以唯一地标识数据块，因此也被称为数据块指纹。数据去重技术使用了 MD5、SHA-1 等安全哈希摘要来唯一标识数据块，即给数据块装置了一个独一无二的指纹，通过这个指纹可以唯一标识数据块，这样简化了重复数据块的识别和匹配的过程：从全局的数据匹配缩小到数据块的哈希匹配。

2.3 指纹索引

索引查找指的是根据数据块的指纹查找匹配，如果有两个数据块的指纹匹配相等，那么它们所表示的数据块也相等。数据存储管理的时候，将非重复数据块直接写入存储系统；如果是重复数据块，则将重复的地址信息记录下来，以便于后续恢复操作。

随着数据规模的持续增长，指纹的数据量会变得异常庞大，所以只能放在磁盘上存储和索引。如果将指纹索引存放在磁盘中会导致查询索引时需要频繁地访问磁盘，系统吞吐量过低。如果只索引部分指纹来减少指纹索引的空间占用又会损失去重率。

假设存在这样一个数据去重系统，其采用了平均 8KB 的数据块大小和 SHA-1 安全哈希算法，这样 80TB 的数据将产生 10M 个数据块、200GB 大小的数据块指纹。内存中显然装不下这 200GB 的指纹，这需要放入磁盘中存储并索引。但是，磁盘访问速度远远低于内存访问速度，这使得数据去重的指纹查找非常缓慢。也就是说，每输入一个数据块，都需要访问磁盘指纹索引，这严重影响了数据去重的系统吞吐率。

随着闪存的发展，为了节省内存成本，大规模的数据去重系统的指纹索引一般放置在磁盘或闪存中^[13]

在使用数据去重技术的备份系统中，去重后的数据块并非直接写入磁盘，而是按序写入内存中的容器。容器是一种定长（例如，4MB）的数据结构，分为两个区域，数据区和元数据区；数据区存放数据块，元数据区存放数据块的指纹、大小和容器内偏移等元数据。当容器的容量达到上限时会一次性写入磁盘。备份系统存储和读写的基本单位是容器。

主流的索引方案可以分为两类：精确去重和近似精确去重。去重过程中，如果指纹在缓存中未命

中则进一步去磁盘中查询全索引,这类方案称为精确去重。如果指纹在缓存中未命中就认为是非重复数据,而不进一步查询全索引,这一类方案称为近似精确去重。近似精确去重允许少量重复数据块被误认为是非重复数据块。现有的索引方案中,采用了抽样的索引方案都是近似精确去重方案。为了减少去重率的损失,指纹的抽样需要利用备份数据流的局部性或相似性。

目前的针对备份的数据去重系统普遍使用了抽样和预取的策略来优化索引查重环节,如 Sparse indexing^[14]、Extreme Binning^[15]和 LIPA^[16]。具体来说,抽样策略指的是从磁盘上的指纹索引中抽样出一部分指纹放进内存中进行索引,从而降低内存开销。预取策略指的是利用局部性和相似性将未来可能重复的指纹预取到内存中。抽样是说处理请求时满足某种条件的指纹才会放到内存索引中,预取是指命中的时候利用局部性进行存储未来可能使用的指纹。

2.4 存储管理

元数据管理与数据管理,与架构相关。

在基于数据去重技术的备份系统中,数据块的数据内容只会保存一份并且被多个数据块所共享。对于非重复数据块,系统会将其数据内容保存在容器中。对于重复数据块,系统仅会记录之前存储过的数据内容所对应的地址信息。这种存储方式有效地提高了存储空间利用率,但是同时也导致了碎片化问题的出现^[17]。碎片化问题指的是单个文件的数据块散落在不同的容器中,这会导致数据恢复时需要更多的磁盘 I/O,降低了数据恢复的效率。

为此,许多数据去重系统采用了重写算法将部分碎片块重新写入磁盘,如 CBR^[18]、Capping^[19]和 HAR^[20]等。CBR 通过对比去重前后的数据流上下文来判断需要重写的碎片块。Capping 限制了数据段的容器引用数量。当数据段的容器引用数量超过阈值时,Capping 会将超出阈值部分的引用率较低的容器进行重写。HAR 利用了碎片块。

具有遗传性这一特点,通过之前备份的历史信息来判定当前备份的碎片块。除了重写外,高效的数据块缓存策略^[21]和新的数据布局方案^[22]也可以提高数据恢复的性能

2.5 在NVM中的应用——Light-Dedup^[28]

(1) 问题背景

非易失性内存(NVM)作为下一代存储介质的应用日益增多,但其高成本制约了发展。通过去除

冗余数据块来降低 NVM 的成本,是一个重要的研究方向。

(2) 现有解决途径

传统的基于磁盘的去重技术,如使用 SHA-256 和 MD-5 等加密哈希算法来识别冗余数据块,并不适合 NVM。因为 NVM 快速的读写能力将性能瓶颈从 I/O 转移到了 CPU。

(3) 不同技术途径优缺点

在探讨非易失性内存(NVM)的去重技术时,应当详细分析 NVM 的 I/O 机制如何影响去重性能。NVM 以其字节可寻址性、持久性和低延迟特性,逐渐成为流行的存储解决方案。然而,高成本仍是其发展的障碍。去重技术能够通过消除冗余数据块来降低 NVM 的成本,但其效率受限于 NVM 特有的 I/O 机制。传统基于磁盘的去重方法,如使用加密哈希算法,已不适应 NVM 的性能特点,因为 NVM 的快速 I/O 能力已将性能瓶颈转移至 CPU。现有的 NVM 去重研究没有充分利用 NVM 的 I/O 性能,缺乏对其 I/O 机制的全面理解。

(4) 对解决问题的价值认识

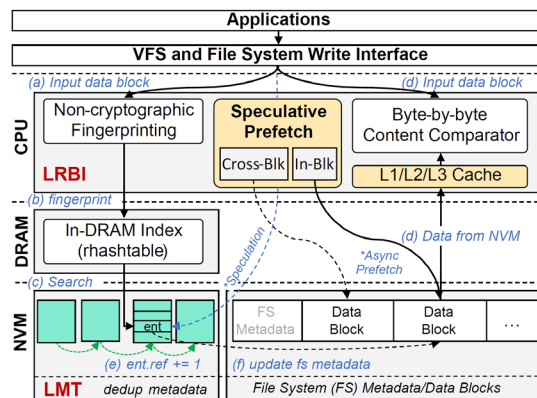


图 2-4 Light-Dedup 架构

Light-Dedup 是为 NVM 文件系统设计的内联去重框架,目的是在不显著增加关键路径开销的情况下,通过考虑 NVM 的内存接口、读/写不对称性和访问粒度,来最大化去重性能。另外,通过组织为基于区域的链接列表,Light-Dedup 能够在文件系统严重老化时(即存在许多空洞时)保持低去重元数据 I/O 放大。

(5) 评述

LRBI 技术:介绍了一种名为 Light-Redundant-Block-Identifier (LRBI)的新技术,Light-Redundant-Block-Identifier(LRBI)结合了非加密哈希和基于推测预取的逐字节内容比较方法,利用 NVM 的内存

接口,通过在 CPU/NVM 缓冲区中推测性预取 NVM 中的数据块,以异步方式读取,显著减少了内容比较的读取延迟。

LMT 设计: 展示了一种名为 Light-Meta-Table (LMT) 的 NVM 内去重元数据表的设计,通过区域为基础的链接列表组织,有效降低了 NVM 粗粒度访问所导致的去重元数据 I/O 放大,同时为了避免垃圾回收的开销, LMT 采用额外的去重元数据空间。

(6) 综合评价

Light-Dedup 的实验结果具有重要意义。该框架在 Linux 内核 5.1.0 上实现,并基于 NOVA,一个现代 NVM 文件系统。结果显示, Light-Dedup 在高重复率下显著提升了文件系统的写入性能,同时几乎不增加任何额外开销。这些结果验证了 Light-Dedup 提出的技术在提高去重性能方面是有效的,特别是对于成本效益更高的 HDD 备份系统。

3 差量压缩技术

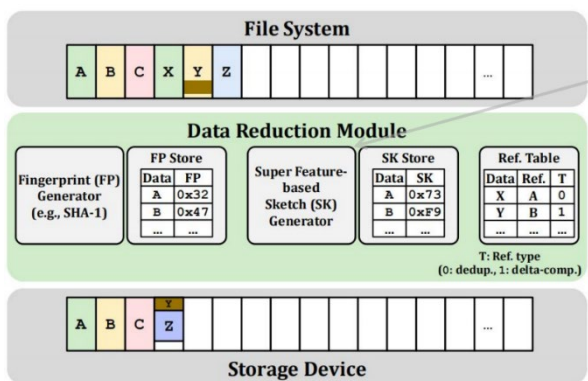


图 3-1 差量压缩结构图

对去重后的数据块进行相似性检测和差量压缩。这是因为数据去重后仍然残留了非重复但很相似的数据,这一类数据将被计算得出截然不同的 SHA-1 指纹,所以无法直接去重。研究人员希望通过差量压缩技术来识别并消除这一类相似数据冗余。这种基于差量压缩的方法较前面两种方案更为复杂,需要首先计算数据块的相似性特征值来检测潜在的差量压缩对象,但这一种方案的数据压缩率收益也是最大的。

差量压缩技术是一种细粒度的无损压缩技术,压缩对象是两个相似的文件或数据块。具体来说,差量压缩会利用滑动窗口技术寻找两个相似对象之间的重复数据。对于重复的数据内容差量压缩会使用 COPY 命令进行编码,仅记录重复部分的位置

和长度信息。对于不重复的数据内容,差量压缩会使用 INSERT 命令进行编码,记录不重复部分的数据内容和长度。以数据块 A 和 B 为例,假设数据块 A 是已经存储过的数据块,而数据块 B 是数据块 A 的相似数据块(通过 sketch 比对),那么差量压缩可以根据数据块 A 的内容对数据块 B 进行差量编码,从而只保存数据块 B 中与数据块 A 不重复的数据内容。在差量编码的过程中,被压缩的数据块被称为目标块,而另一个完全存储的数据块被称为引用块。

由于差量压缩可以有效地消除相似对象之间的冗余数据,增强数据去重的冗余数据删除效果,因此二者常常联合起来进行冗余数据删除,典型的相关研究包括 TAPER^[23]和 SIDC^[6]等。TAPER 联合了数据去重和差量压缩技术来对文件进行多级压缩,先做相似性检测,再在相似文件中进行细粒度的块级(block-level)去重;最后,剩下的非重复数据再使用差量压缩进一步减少数据冗余。其中粗粒度的块级去重平均块长为 4KB,而细粒度的块级去重块长为 700 字节。SIDC 在现有的数据去重框架上提出了一种基于数据流局部性的差量压缩策略。

3.1 相似性检测技术

相似性检测技术目的在于快速判断两个对象或数据序列是否相似,主流的相似性检测技术包括:

3.1.1 近似指纹 (Approximate Fingerprints) ^{[24][25]}

由于相似文件的指纹同样具有相似性, Approximate Fingerprints 采用抽样的特征提取策略从文件的所有子字符串的 Rabin 指纹集合中选出一部分相似指纹。这个过程类似于数据分块算法中的切点判断。具体来说, Approximate Fingerprints 将文件分成多个可重叠的定长连续字符序列(也被称为(shingle)。将这种定长连续字符序列称为数据片段。如果文件的某个数据片段的 Rabin 指纹 fp 满足条件: $fp \% D = R$, 则 fp 被选为文件的特征值。其中 D 表示指纹抽样率, R 是算法的预设值。在特征提取步骤结束后, Approximate Fingerprints 会根据文件之间共享的相似指纹的数量来判断文件的相似度。然而, Approximate Fingerprints 提取出的特征值的数量取决于文件的大小和指纹抽样率。这导致了 Approximate Fingerprints 需要花费较大的匹配开销才能完成文件之间的相似度判断。简单来说,该方法通过抽样和分块,计算两个文件

之间相同块的个数判断相似度，但是两个文件之间特征匹配的开销大。。

3.1.2 局部敏感哈希 (LSH)

局部敏感哈希 (Locality sensitive hashing, 简称 LSH) 能对相似的数据内容生成相似的指纹。LSH 评估相似度的准确度取决于采用的哈希方式，最常用的哈希方式包括 MinHas 和 SimHash。

(1) MinHash^[26]

MinHash 基于二元向量的相似度评估方法用集合的 Jaccard 距离来描述相似度。

(2) SimHash^[27]

SimHash 是 Charikar 与 2002 年提出的另一种 LSH。SimHash 基于的是余弦相似度，是一种降维方法，能将文件映射成少量位数的指纹。具体来说，SimHash 首先将文件转换成一个由高维度向量组成的带有权重的特征集合，然后将高维度向量转换成小容量的指纹。特征匹配时，SimHash 通过计算两个指纹间不同的二进制值的数量来估计相似度；不相同的值的数量（即距离）越少表示相似度越高。SimHash 常用于检测相似网页。

3.1.3 Super-feature

其是 Broder 提出的一种基于相似性特征值的相似性检测方法。

不同于 Approximate Fingerprints, Super-feature 确保了每个数据块提取出的特征值数量是相同的。Super-feature 对数据块的所有数据片段的 Rabin 指纹计算哈希函数，并从哈希值集合中选出数据块的特征值。在实际应用中，Super-feature 使用线性变换的方式来代替哈希函数。在特征提取步骤结束后，Super-feature 会将固定数量的特征值打包成多个超级特征值。如果两个数据块的某个超级特征值相同，则表示它们有很大的概率是相似的。Super-feature 的超级特征值策略将特征匹配过程简化为查询超级特征值索引，极大地降低了相似性检测的检测开销。

由于较好的相似数据检测效果，Super-feature 被广泛地应用在了存储系统中。然而，Super-feature 的特征提取步骤，从数据块中提取 N 个特征值意味着数据块的每个数据片段所对应的指纹都需要进行 N 次乘法、N 次加法、N 次求模、N 次数值比较以及 N 次条件判断操作，需要对每个数据块的所有数据片段的 Rabin 指纹进行多次线性变换操作，具有很大的计算开销。这导致了 Super-feature 的特征提取速率很慢，影响了冗余数据删除系统的

整体性能。

$$Feature_i = \text{Max}_{j=1}^{\text{len}} \{ (m_i * fp_j + a_i) \bmod 2^{32} \}$$

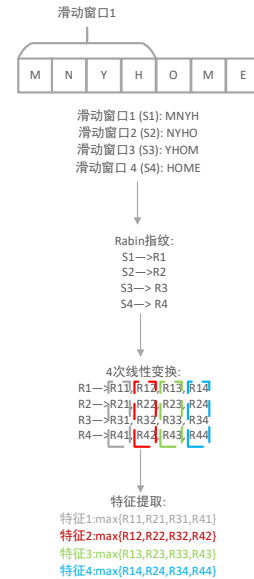


图 3-2 Super-Feature 流程

3.1.4 Finesse

Finesse 是 Zhang 等人提出的一种基于特征局部性的相似性检测方法。Finesse 将数据块各个子区域的最大 Rabin 指纹作为数据块的特征值，避免了复杂的线性变换操作。这种新的特征提取策略有效地降低了相似性检测的计算开销，提高了数据块特征提取的速率。然而，Finesse 的子区域划分操作会带来特征值偏移问题，造成部分相似数据块无法被检测出，降低了冗余数据删除系统的去重效果。在 fast22 deepsketch 论文中将 Finesse 作为 baseline，对其评价是 the state-of-the-art SF-based technique。

动机：从子块中计算特征的动机是，观察到细粒度的局部性广泛存在于检测到的相似块中。下图提供了这种局部性的一个例子：单个块的子区域（子块）也以相同的顺序出现在它们高度相似的块中，且概率非常高，这意味着这些子块彼此之间也非常相似。

3.1.5 数据特征化的 Rabin 指纹技术

特征提取中的数据特征化指的是将文件或数据块的数据内容转化为哈希值集合的过程。根据特征化粒度的不同，相似性检测方法可以分为粗粒度的相似性检测方法和细粒度的相似性检测方法。

粗粒度的相似性检测方法一般以数据块作为基本单位，将文件的数据内容转化为数据块指纹的集合。这种基于数据块的粗粒度相似性检测方法的计算开销较小，可以用于优化数据去重系统的去重效果或实现基于相似度的冗余数据删除系统，如

dbDedup 等。然而，这种相似性检测方法只能粗略地判断文件或数据段之间的相似度，具有较低的准确度。细粒度的相似性检测方法一般将数据片段作为基本单位，将文件或数据块的数据内容转化为数据片段的哈希指纹集合。这种相似性检测方法的计算开销较大，但是具有较好的相似数据检测效果，常常被用于联合数据去重和差量压缩的冗余数据删除系统中。目前最常用的 Super-feature 就是典型的基于数据片段的细粒度相似性检测方法。



图 3-3 Rabin 滑动窗口

$$\begin{aligned} Rabin(B_i, \dots, B_{i+m-1}) \\ = [Rabin(B_{i-1}, \dots, B_{i+m-2}) \\ - B_{i-1}c^{m-1}]c + B_{i+m-1} \end{aligned}$$

在基于数据片段的细粒度相似性检测方法中，最常用的数据特征化算法是基于 Rabin 指纹的滑动哈希算法。Rabin 滑动哈希算法会将数据块划分成多个数据片段并生成所有数据片段的 Rabin 指纹集合。具体来说，Rabin 滑动哈希算法使用定长的滑动窗口对数据块的内容进行扫描。滑动窗口最开始位于数据块的起始位置，之后逐字节滑动至数据块的尾部。滑动窗口每滑动一次就计算一次窗口内字节序列的 Rabin 指纹，其中 m 是滑动窗口的大小，一般设置为 32 或 48 字节。 B_1, B_2, \dots, B_w 表示滑动窗口中长度为 w 的字节序列， c 代表预设的常量。

3.1.6 特征选取

在 Rabin 指纹算法为数据块的每个数据片段计算 Rabin 指纹的同时，相似性检测方法还会对生成的 Rabin 指纹执行相应的操作，从而选出能够代表数据块内容的特征值。这一过程指的就是特征提取过程中的特征值选取阶段。不同的相似性检测方法采用了不同的方法选取特征值。

经典的相似性检测方法 Super-feature 采用线性变换的方式来生成数据块的特征值。具体来说，Super-feature 将数据块的所有数据片段的 Rabin 指纹进行 N 次线性变换操作，并将每个线性变换操作所对应的最大的哈希值被选为数据块的特征值。

Finesse 指出相似数据块之间存在细粒度的特征局部性，即相似数据块内部的特征值一般会以相

同的顺序出现。基于这个观察，Finesse 首先将数据块按照位置划分成 N 个大小相等的子区域，然后将所有数据片段的 Rabin 指纹集合按照子区域的位置分成 N 个大小相等的 Rabin 指纹子集合，最后将每个子集合中最大 Rabin 指纹选为数据块的特征值。

3.1.7 DeepSketch^[9]

DeepSketch 提出了一种新的用于重复数据消除后增量压缩的参考搜索技术。

(1) 问题背景

在数据中心，每天产生的海量新数据对经济型存储解决方案的需求日益增长。数据去重技术减少存储介质中物理存储的数据量，通过减少数据冗余来降低数据中心的管理成本。这允许数据中心用更少或更小的资源（例如存储设备和服务器）处理相同量的数据。现有的重复数据删除后增量压缩技术提供与最佳数据缩减率相比，数据缩减率明显较低。主要是由于增量压缩的参考搜索精度有限，无法为许多传入的数据块识别出良好的参考块。

(2) 现有解决途径

综合数据去重和无损压缩技术，数据去重首先应用于即将存储的数据块，随后对未去重的数据块执行无损压缩。

(3) 技术途径的优缺点

尽管数据去重和无损压缩的结合使用在减少数据冗余方面取得了一定成果，但现有技术在使用局部敏感哈希 (LSH) 基准搜索时，高误报率 (FNR) 的问题导致其在数据去重效率上远低于最优可能性。

(4) 研究的重要性

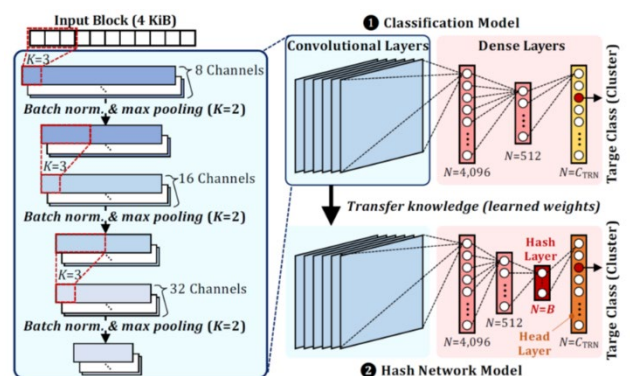


图 3-4 学习到哈希方法

DeepSketch 通过学习到哈希方法，利用深度神经网络 (DNN) 提取数据块的草图，以保持与其他块的相似性，提高去重后增量压缩的参考搜

索精度，从而提高数据去重效率，特别是在数据中心的存储系统对空间效率的优化至关重要。

(5) 评述

DeepSketch 利用深度神经网络 (DNN) 模型生成每个数据块的草图。该方法面临的挑战包括处理高维数据集和缺乏明确的标签数据或语义信息。为此, DeepSketch 开发了一种新的训练方法, 包括动态 k-均值聚类 (DK-Clustering), 通过随机修改数据块来防止 DNN 训练的偏差, 以及两阶段 DNN 训练, 从而产生数据块的哈希值。

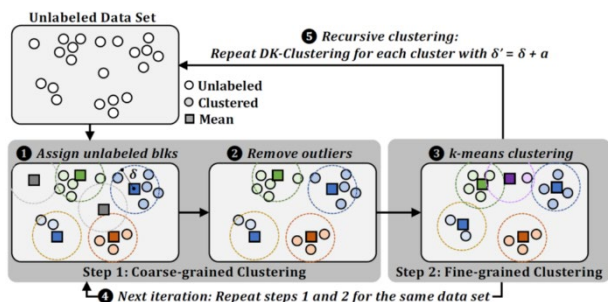


图 3-5 动态 k-均值聚类 (DK-Clustering)

(6) 总结

DeepSketch 的关键贡献在于提出了一个基于机器学习的去重后增量压缩参考搜索技术, 并通过新的训练方法和近似最近邻搜索来改进数据去重率, 其评估结果显示, 在十一种真实世界的工作负载中, DeepSketch 将数据去重率提高了最多 33% (平均 21%)。

DeepSketch 使用深度神经网络提取数据块的 sketch, 利用深度学习的方法来实现增量压缩, 获得更高的搜索精度, 从而提高数据缩减效率。

3.1.8 LoopDelta^[29]

(1) 问题背景

数据去重是计算系统中常用来提高存储空间和网络带宽效率的技术, 但这种技术通常无法消除非重复但高度相似的数据块之间的冗余。Delta 压缩就是用来进一步从非重复但相似的数据块中移除冗余数据的技术。

(2) 现有解决途径

传统的后去重增量压缩方法在备份存储中要么因为检测到的块之间相似度低而效率不高, 要么因为为读取基块增加额外 I/O 操作而降低了 (备份和恢复) 吞吐量, 或者为备份系统增加了额外的服务中断操作。

(3) 不同技术途径优缺点

现有的基于逻辑-局部性的索引技术和基于物理-局部性的索引技术在检测相似块方面有互补的能力, 但是它们各自都有缺陷。前者可能错过一些潜在的相似块, 而后者则可能因为检测到的块相似度低而效率不高。

(4) 对解决问题的价值认识

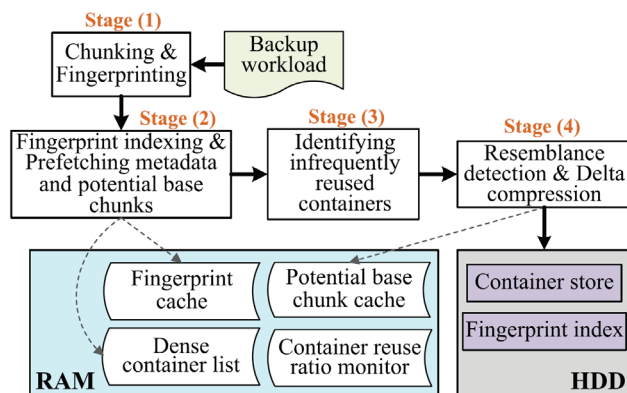


图 3-6 LoopDelta 结构

LoopDelta 是一个增强的嵌入式去重增量压缩方案, 它非侵入式地结合了四种关键技术: 双重局部性相似度跟踪、局部性感知预取、缓存感知过滤器和逆向增量压缩, 以提高压缩比, 改善备份吞吐量和恢复性能。这是一种结合了数据去重和增量压缩的先进技术。其核心在于利用数据块之间的逻辑和物理局部性来提高压缩比和恢复性能。LoopDelta 通过跟踪数据块和基块, 并预取所需的基块, 实现了高效的数据压缩, 特别是在非重复但高度相似的数据块之间。

(5) 评述

LoopDelta 提出了四种关键技术来解决现有去重增量压缩方法的局限性: 双重局部性相似度跟踪 (利用逻辑和物理局部性来侦测潜在相似的块)、局部性感知预取 (通过日常操作期间预取元数据来避免额外的 I/O 操作)、缓存感知过滤器 (在恢复过程中避免需要额外 I/O 操作的基块的引用) 和逆向增量压缩 (对那些通常不作为基块的数据块进行增量压缩)。这些技术共同作用, 显著提升了数据去重和压缩的整体效率。具体如下:

双重局部性相似度跟踪: 这一技术结合了逻辑和物理局部性来侦测潜在相似的块。逻辑局部性是指相邻备份之间的数据块具有高度的相似性, 而物理局部性是基于存储单元 (如容器) 中保留的相似性。通过利用这两种局部性, LoopDelta 能够有效地跟踪和识别出在连续备份中可能相似的数据块对。

只需对少量特征值索引做预取，避免了建立全局特征值索引以及对应的内存开销（去重的预取容器里的 SF）。Dare 利用数据流的局部性对增量压缩算法进行了优化（相邻作为相似）。

3.2 增量编码

增量压缩算法所完成的工作是对被判断为相似的两个数据块采用增量编码算法来缩减数据，消除数据块内部的重复信息。

Xdelta 和 Zdelta 是目前存储系统中最常用的增量压缩算法。

Xdelta 采用了一种基于滚动窗口的技术来产生字符串，这些字符串将被用做相似数据块之间冗余数据识别的最小单位。为了最大限度的寻找重复字符串，Xdelta 采用了逐字节滑动字符串窗口的方法。这样即使两个相似数据块之间的内容频繁地存在内容差异，这种滚动窗口技术还是能够足够精细地发现细粒度的重复字符串。其计算开销和索引开销较大。

由于增量编码的过程具有较大的计算开销，Xia 等人提出了 Ddelta 和 Edelta。Ddelta 和 Edelta 使用了数据去重的思想来快速查找相似数据块内部的重复字符串，大大加快了增量编码的过程。

Ddelta 增量编码的主要方法包括有：

（1）采用了数据去重中基于内容分块和指纹识别的思想来简化和加速增量编码的增量计算比较过程；

（2）针对基于内容分块算法导致的增量压缩率下降的问题，利用冗余数据的局部性来补充查找可能的更多的冗余数据。

然而，这种粗粒度的字符匹配方式会降低增量编码的压缩效果。为此，Tan 等人提出了 Gdelta。Gdelta 不仅使用了基于 Gear 的快速滑动哈希算法和基于数组的索引来加速编码过程，同时还利用了压缩工具 Zstd 和 FSE 进一步压缩增量编码后的增量数据，取得了较好的数据压缩效果。

3.3 存储管理

存储管理用于持久化容器、文件谱、指纹索引、布隆过滤器等信息。

（1）容器。容器包含了经过去重和压缩后的数据块的数据内容以及它们的元数据信息。可以通过读取容器查找出之前存储过的数据块。

（2）文件谱。文件谱记录着文件的所有指纹序列和指纹对应的数据块的元数据信息。可以通过读取文件谱和相应的容器来进行数据恢复。

（3）元数据缓存。元数据缓存包含了最近访问的多个容器的元数据区数据。数据块的超级特征值和指纹共同保存在容器的元数据区中。可以通过查询元数据缓存来判断数据块是否重复和相似。为了方便查找潜在的相似数据块，元数据缓存中每个容器中的数据块元数据会以双链表的形式保存。

（4）指纹索引。指纹索引记录着所有之前存储过的指纹和指纹对应的数据块所在的容器的编号。

（5）布隆过滤器。当元数据缓存中没有找到数据块的指纹时，可以通过布隆过滤器判断该指纹是否可能存在于指纹索引中。

（6）容器缓存。容器缓存包含了最近访问的容器，用于减少特征匹配过程对磁盘的随机访问。

4 总结与展望

本文以冗余数据删除技术作为研究对象，主要探讨了数据去重、增量压缩和相似性检测算法，以提高数据压缩效率和性能。

本文认为目前仍有许多问题有待进一步探讨和解决。在今后的研究中，希望能够从以下几个方面进行深入研究：

针对动态数据集的压缩问题，希望能够设计一种能够实时更新和增量压缩的方法，以适应数据集的变化情况，提高压缩效率和灵活性。

针对不同类型的并行平台的压缩问题，希望能够探索一种能够兼容多种平台的压缩方法，以充分利用各种平台的并行特性，提高压缩性能和可移植性。

针对数据压缩效率和性能的优化问题，希望能够改进现有的相似性检测算法和增量压缩算法，以进一步降低复写率、提高压缩比、降低时间开销等指标，提高压缩质量和速度。

相信随着技术的不断发展和创新，冗余数据删除技术将会有更广阔的应用前景和更大的发展潜力。

致谢 感谢施老师、童老师和胡老师这段时间为我们上课作出的充分准备和付出的辛勤劳动。老师上课条理清楚，重点突出，对重难点问题的解释深入浅出，通俗易懂，并且为我们精心准备了 2023 年

顶会论文题目, 让我们有的放矢, 更好地磨炼自己。同时老师对同学们提出的问题耐心解释, 也能够充分听取同学们对这门课的意见, 经过这段时间的学习, 我收获颇多, 再次感谢老师的指导, 也感谢我的家人和同学对我提供的帮助!

参考文献

- [1] Shilane P, Wallace G, Huang M, et al. Delta Compressed and Deduplicated Storage Using Stream-Informed Locality[C]//HotStorage. 2012.
- [2] Huffman D A. A method for the construction of minimum-redundancy codes[J]. Proceedings of the IRE, 1952, 40(9): 1098-1101..
- [3] Ziv J, Lempel A. A universal algorithm for sequential data compression[J]. IEEE Transactions on information theory, 1977, 23(3): 337-343.
- [4] Ziv J, Lempel A. Compression of individual sequences via variable-rate coding[J]. IEEE transactions on Information Theory, 1978, 24(5): 530-536.
- [5] Broder A Z. Identifying and filtering near-duplicate documents[C]//Combinatorial Pattern Matching: 11th Annual Symposium, CPM 2000 Montreal, Canada, June 21–23, 2000 Proceedings 11. Springer Berlin Heidelberg, 2000: 1-10.
- [6] Shilane P, Huang M, Wallace G, et al. Wan-optimized replication of backup datasets using stream-informed delta compression[J]. ACM Transactions on Storage (ToS), 2012, 8(4): 1-26.
- [7] Xia W, Jiang H, Feng D, et al. DARE: A deduplication-aware resemblance detection and elimination scheme for data reduction with low overheads[J]. IEEE Transactions on Computers, 2015, 65(6): 1692-1705.
- [8] Park J, Lee S, Kim J. DAC: Dedup-assisted compression scheme for improving lifetime of NAND storage systems[C]//Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. IEEE, 2017: 1249-1252.
- [9] Park J, Kim J, Kim Y, et al. DeepSketch: A New Machine Learning-Based Reference Search Technique for Post-Deduplication Delta Compression[C]//20th USENIX Conference on File and Storage Technologies (FAST 22). 2022: 247-264
- [10] Spring N T, Wetherall D. A protocol-independent technique for eliminating redundant network traffic[C]//Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. 2000: 87-95.
- [11] Bjørner N, Blass A, Gurevich Y. Content-dependent chunking for differential compression, the local maximum approach[J]. Journal of Computer and System Sciences, 2010, 76(3-4): 154-203.
- [12] Zhang Y, Jiang H, Feng D, et al. AE: An asymmetric extremum content defined chunking algorithm for fast and bandwidth-efficient data deduplication[C]//2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015: 1337-1345.
- [13] Wang Q, Li J, Xia W, et al. Austere Flash Caching with Deduplication and Compression[C]//USENIX Annual Technical Conference. 2020: 713-726.
- [14] Lillibridge M, Eshghi K, Bhagwat D, et al. Sparse indexing: Large scale, inline deduplication using sampling and locality[C]//Fast. 2009, 9: 111-123.
- [15] Bhagwat D, Eshghi K, Long D D E, et al. Extreme binning: Scalable, parallel deduplication for chunk-based file backup[C]//2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems. IEEE, 2009: 1-9.
- [16] Xu G, Tang B, Lu H, et al. Lipa: A learning-based indexing and prefetching approach for data deduplication[C]//2019 35th Symposium on mass storage systems and technologies (MSST). IEEE, 2019: 299-310.
- [17] 付恣. 面向数据备份的高效数据去重系统构建方法研究[D]. 华中科技大学博士学位论文, 2016.
- [18] Kaczmarczyk M, Barczynski M, Kilian W, et al. Reducing impact of data fragmentation caused by in-line deduplication[C]//Proceedings of the 5th Annual International Systems and Storage Conference. 2012: 1-12.
- [19] Lillibridge M, Eshghi K, Bhagwat D. Improving Restore Speed for Backup Systems that Use Inline Chunk-Based Deduplication[C]//11th USENIX Conference on File and Storage Technologies (FAST 13). 2013: 183-197.
- [20] Fu M, Feng D, Hua Y, et al. Accelerating Restore and Garbage Collection in Deduplication-based Backup Systems via Exploiting Historical Information[C]//USENIX Annual Technical Conference. 2014: 181-192.
- [21] Cao Z, Wen H, Wu F, et al. ALACC: Accelerating restore performance of data deduplication systems using adaptive look-ahead window assisted chunk caching[C]//16th USENIX Conference on File and Storage Technologies (FAST 18). 2018: 309-324.
- [22] Zou X, Yuan J, Shilane P, et al. The dilemma between deduplication and locality: Can both be achieved?[C]//FAST. 2021: 171-185.
- [23] Jain N, Dahlin M, Tewari R. TAPER: Tiered Approach for Eliminating Redundancy in Replica Synchronization[C]//Fast. 2005, 5: 21-21.
- [24] Manber U. Finding similar files in a large file system. In: Proceedings of the USENIX Winter 1994 Technical Conference[J]. 1994.
- [25] Douglass F, Iyengar A. Application-specific Delta-encoding via Resemblance Detection[C]//USENIX annual technical conference, general track. 2003: 113-126.
- [26] Broder A Z, Charikar M, Frieze A M, et al. Min-wise independent permutations[C]//Proceedings of the thirtieth annual ACM symposium on Theory of computing. 1998: 327-336.

- [27] Charikar M S. Similarity estimation techniques from rounding algorithms[C]//Proceedings of the thirty-fourth annual ACM symposium on Theory of computing. 2002: 380-388.
- [28] Qiu J, Pan Y, Xia W, et al. Light-Dedup: A Light-weight Inline Deduplication Framework for Non-Volatile Memory File Systems[C]//2023 USENIX Annual Technical Conference (USENIX ATC 23). 2023: 101-116.
- [29] Zhang Y, Jiang H, Feng D, et al. LoopDelta: Embedding Locality-aware Opportunistic Delta Compression in Inline Deduplication for Highly Efficient Data Reduction[C]//2023 USENIX Annual Technical Conference (USENIX ATC 23). 2023: 133-148.
- [30] Zhang Y, Jiang H, Wang C, et al. Applying Delta Compression to Packed Datasets for Efficient Data Reduction[J]. IEEE Transactions on Computers, 2023.

附录：汇报记录。

1. 相似度的度量指标是什么？增量编码的相似块是如何存储的？

(1) 度量指标增量压缩效率 (Delta Compression Efficiency, 简称DCE) 由FAST19 的Finesse论文更早提出, DCE用于估计相似性检测算法检测到的相似块之间的相似度, 即增量压缩后的块数据大小与增量压缩前的块数据大小之比。DCE强调的是检测到的相似块本身。因此, 更高的DCE意味着检测相似块的误报概率更低。

$$DCE = \frac{\text{增量压缩后的块数据大小}}{\text{增量压缩前的块数据大小}}$$

(2) 增量编码的相似块存储主要通过以下结构完成:

①容器。容器包含了经过去重和压缩后的数据块的数据内容以及它们的元数据信息。可以通过读取容器查找出之前存储过的数据块。

②文件谱。文件谱记录着文件的所有指纹序列和指纹对应的数据块的元数据信息。可以通过读取文件谱和相应的容器来进行数据恢复。

③元数据缓存。元数据缓存包含了最近访问的多个容器的元数据区数据。数据块的超级特征值和指纹共同保存在容器的元数据区中。可以通过查询元数据缓存来判断数据块是否重复和相似。为了方便查找潜在的相似数据块, 元数据缓存中每个容器中的数据块元数据会以双链表的形式保存。

④指纹索引。指纹索引记录着所有之前存储过的指纹和指纹对应的数据块所在的容器的编号。

⑤布隆过滤器。当元数据缓存中没有找到数据块的指纹时, 可以通过布隆过滤器判断该指纹是否可能存在于指纹索引中。

⑥ 容器缓存。容器缓存包含了最近访问的容器, 用于减少特征匹配过程对磁盘的随机访问。

2. LoopDelta使用基于双局部性的草图索引的原因动机是什么？什么样的数据集是既具有逻辑位置局部性又具有物理位置局部性？

(1) LoopDelta 使用基于双局部性的草图索引的原因动机源于对现有数据去重和增量压缩技术的限制的观察和分析。这种方法结合了逻辑位置局部性和物理位置局部性的优势, 以提高相似块的检测效率和数据压缩率。具有逻辑位置局部性的数据集指的是在连续备份中显示重复模式的数据, 这种重复模式在去重之前在备份流中保持。而具有物理位置局部性的数据集是在去重后, 数据块在物理存储单元 (例如容器) 中保持的重复模式。LoopDelta 的设计旨在通过同时利用这两种局部性来优化相似块的检测, 并通过减少额外的 I/O 操作和避免在备份过程中的服务中断, 从而提高备份和恢复的效率。

(2) 具有逻辑位置局部性和物理位置局部性的数据集在生活中, 一个典型的具有逻辑位置局部性和物理位置局部性的数据集的例子可以是个人电脑上的文件系统。考虑这样一个场景:

逻辑位置局部性: 假设你经常编辑和保存一个工作文档。每次你进行了少量更改并保存, 你的备份系统会创建这个文件的新版本。在这种情况下, 这个文件的多个版本之间会有许多共同的数据块, 因为大部分内容在每个版本中都是相同的。这些共同的数据块反映了逻辑位置局部性, 即在连续备份中显示出的重复模式。

物理位置局部性: 同时, 备份系统可能会将这些文件的所有版本存储在同一个物理硬盘驱动器或网络存储上。即使文件内容在不同版本之间略有变化, 这些变化的版本 (数据块) 仍然存储在相同或邻近的物理位置上。这种情况下, 数据块在物理存储单元中的聚集就反映了物理位置局部性。

在这个例子中, 通过识别这些逻辑上和物理上接近的数据块, 一个智能的备份系统可以更有效地进行数据去重和增量压缩, 从而节省存储空间并提高备份和恢复的效率。

更广泛地, 图像和视频编辑中也存在着冗余相似数据, 如视频编辑中的相似帧。但是, 一般意义上的数据去重和增量压缩如LoopDelta针对的是一维字节流, 导致其对二维的图像、三维的视频冗余数据删除效果不佳, 一般常

用于通用文件系统备份数据的缩减。

不过，在二维图像领域冗余数据删除技术也开辟了新战场，2022 年有学者提出了 imDedup 系统用于二维图像的数据去重和差量压缩：

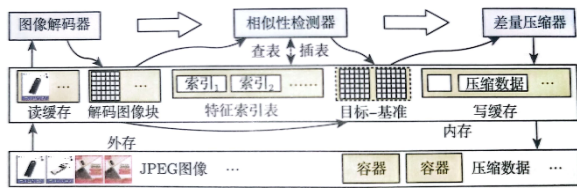


图 4-1 imdedup 架构

imDedup 是一种用于图像去重的框架，它通过解码图像以暴露图像间的相似性，然后利用细粒度的压缩方法消除相似图像对之间的冗余数据，最后重新压缩非冗余数据。

imDedup 包括三个主要模块：JPEG 解码器、相似性检测器和 Idelta 编码器。

① JPEG 图像解码器将 JPEG 图像解码为 DCT 块（一个短整型二维数组），以便于检测相似性，因为相同的像素块在经过相同的 JPEG 编码器编码后所得到的 DCT 块也是一样的，使得 DCT 块保持了图像压缩前的相似性，并且，DCT 块和 JPEG 图像之间的转换是可逆的，保证了后续过程的无损性；

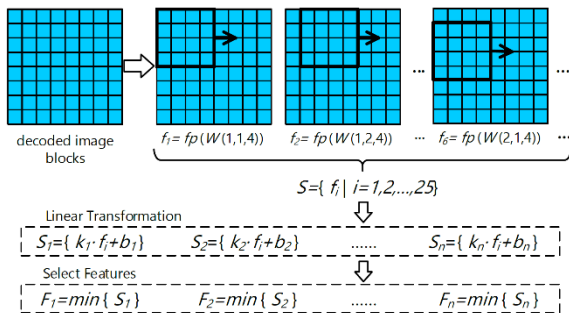


图 4-2 imDedup 图像二维特征提取

② 相似性检测器用于提取图像特征并寻找最相似的图像；

③ Idelta 差量压缩器则负责压缩图像，并在块单位上消除冗余，从而保证剩余数据仍然保持块结构并适合 JPEG 格式，从而提高压缩比和吞吐量。

此外，imDedup 的实现还包括基于特征图（Feature Map）的高效相似性检测方法和面向 JPEG 数据的定制化 Idelta 压缩方法，优化了处理速度和压缩效果。

当然，现在针对三维视频的成熟冗余数据删除技术仍未出现，这也是未来冗余数据删除技术发展的一大方向。