

# 对重复数据消除的过去、现在和未来的全面研究

夏雯, 成员, IEEE, 洪江, 研究员, IEEE, 冯丹, 成员, IEEE, Fred Douglass, 高级成员, IEEE, Philip Shilane, 余华, 高级成员, IEEE, 傅敏, 张玉成和周玉坤

摘要—由于数字数据的爆炸式增长, 重复数据删除作为一种高效的数据缩减方法, 在大规模存储系统中日益受到关注和普及。它消除了文件或子文件级别的冗余数据, 并通过其加密安全的哈希签名(即防冲突指纹)来识别重复内容, 该签名在计算上比大规模存储系统中的传统压缩方法高效得多。本文首先回顾了重复数据消除的背景和关键特征, 然后根据重复数据消除过程的关键工作流程, 对重复数据消除的研究现状进行了总结和分类。重复数据消除技术的总结和分类有助于识别和理解重复数据消除系统最重要的设计考虑事项。此外, 我们还讨论了重复数据删除的主要应用和行业趋势; 并提供重复数据消除研究和调查的公开来源列表。最后, 我们概述了基于重复数据消除的存储系统面临的开放问题和未来的研究方向。

索引术语—重复数据消除、数据压缩、存储系统、存储安全、数据缩减、增量压缩

## 1 介绍

世界上的数字数据量呈爆炸式增长, 部分证据是 2010 年和 2011 年生成的估计数据量分别从 1.2 兆字节大幅增加到 1.8 兆字节[1, 2], 预计 2020 年产生的数据量为 44 兆字节[3, 4]。由于这种“数据洪流”, 如何经济高效地管理存储已成为大数据时代海量存储系统中最具挑战性和最重要的任务之一。微软进行的工作量研究[5, 6]和 EMC [7, 8]表明其生产主存储系统和辅助存储系统中分别约有 50% 和 85% 的数据是冗余的。根据国际数据公司最近的一项研究[9], 近 80% 的受访企业表示, 他们正在探索存储系统中的数据重复数据消除技术, 以减少冗余数据, 从而提高存储效率并降低存储成本。

重复数据消除是一种高效的数据减少方法, 不仅减少了存储空间[5-7, 10-13]通过消除重复数据, 同时最大限度地减少冗余数据在低带宽中的传输

网络环境[8, 14, 15]。一般来说, 典型的块级重复数据消除系统分割输入数据流(例如, 备份文件、数据库快照、虚拟机映像等。)分成多个数据“块”, 每个数据“块”由密码安全散列签名(例如, SHA-1, 也称为指纹[1])唯一识别和重复检测[11, 14]。这些块的大小可以固定[11], 如文件块, 或由内容本身决定的可变大小的单元[14]。然后, 重复数据消除系统删除重复的数据块, 只存储或传输其中的一个副本, 以实现节省存储空间或网络带宽的目标。

传统上, 数据缩减是数据压缩方法的结果, 这些方法使用字典模型来识别短字符串(例如 16 字节)的冗余, 例如经典的 LZ77/LZ88 算法[16, 17]。大多数方法首先计算字符串的弱散列, 然后逐字节比较散列匹配的字符串。由于时间和空间的复杂性, 基于字典模型的压缩方法, 如 LZ0 [18]LZW[19], DELTE[20], 仅压缩小得多的区域中的数据, 例如文件或一组小文件中的数据[21], 这在处理速度和压缩效率之间进行了权衡。

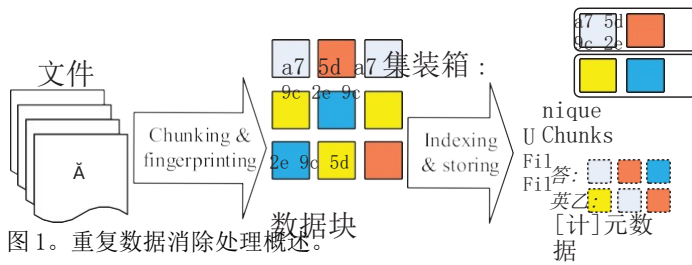
对于大规模存储系统, 重复数据删除比传统的压缩方法(如霍夫曼编码或 LZ 压缩)更具可扩展性和效率。重复数据消除的主要优势有两方面。首先, 重复数据消除在区块(例如 8KB)或文件级别识别和消除冗余, 而传统的压缩方法在字符串或字节级别工作。其次, 重复数据消除通过计算其加密安全的基于哈希的指纹(在本文的剩余部分简称为“指纹”)来识别重复内容(文件或块),

W. 夏, 华中科技大学武汉光电国家实验室计算机科学与技术学院, 430074。电子邮件: xia@hust.edu.cn.

H. 姜就职于美国得克萨斯州阿灵顿市大道 500 号 640 号得克萨斯大学计算机科学与工程系。电子邮件: hong.jiang@uta.edu.

F. Douglass 和 P. Shilane 在美国新泽西州普林斯顿的 EMC 公司工作。电子邮件: fred.douglass, philip.shilane @emc.com.

D. 冯, 华, 付, 张, 周, 武汉大学计算机学院光电国家实验室和科技, 华中科技大学, 430074 中国。对应的作者是冯婷。电子邮件: dfeng、基督教青年会、富民、基督教青年会、ykzhou @hust.edu.cn.

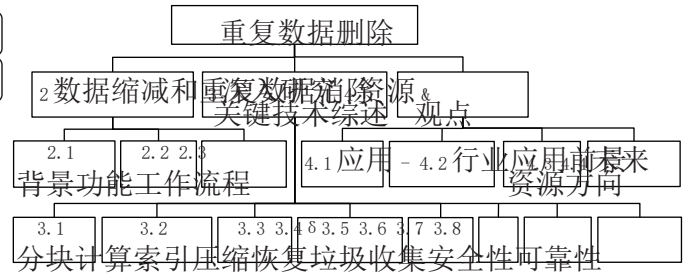


这避免了传统的逐字节比较方法。这两项功能使重复数据消除能够通过计算和索引区块或文件的指纹，轻松适用于大规模存储系统中的全局数据缩减。这是因为重复数据消除的指纹大小比原始数据小几个数量级。

通常，大多数块级重复数据消除方法的主要工作流程包括五个关键阶段，即分块、指纹识别、指纹索引、进一步压缩和存储管理。进一步的压缩是可选的，包括非重复组块的传统压缩（例如，LZ 压缩）和非重复但相似组块的增量压缩。重复数据消除系统中的存储管理可以分为几类，如数据恢复（碎片消除）、垃圾收集、可靠性、安全性等。

数字 1 显示了数据重复数据消除的一般工作流程。文件首先被分成大小相等或相似的块，每个块由它的指纹唯一地表示。重复数据消除仅通过使用指纹（索引）快速验证唯一性，将唯一（即非重复）区块存储在磁盘上。它还在元数据中记录将用于重建原始文件的组成块的列表。针对大规模存储系统中指纹大小溢出内存容量的问题，提出了几种加速磁盘索引查找过程的优化方法。一个典型的例子是 Data Domain 文件系统 (DDFS) [13] 使用布隆过滤器 [22] 和一个局部性保留的缓存，用于加速重复数据消除的指纹索引，详见第 3.3 节。通常，唯一的块将被存储到几个称为容器的大型固定大小的存储单元中 [13]，因此每个文件的恢复将由于块碎片而导致容器的许多随机输入/输出 [23]：重复数据消除后，文件的块会分散在不同的容器中。区块和元数据的详细存储管理将在第 3 节中讨论。此外，还有许多公开的重复数据消除教程 [24 - 27]，这可能也有助于理解重复数据消除概念。

在本文中，我们将研究的重点放在数据重复数据消除工作流程的五个阶段中的每一个阶段，以深入了解这些年来该技术的发展以及这些最新方法的优缺点，并概述尚未解决的问题。



基于重复数据消除的存储系统面临的问题和研究挑战。我们认可并赞赏最近几项重复数据消除调查研究的贡献 [28 - 32] 介绍了重复数据消除策略和使用案例，重点介绍了重复数据消除的特定方面（例如，索引方案 [31]），或调查重复数据消除技术在特定领域（例如云存储 [32]）。然而，我们的研究不同于这些先前的调查，因为我们研究冗余数据减少技术的发展和关键特征。更重要的是，我们深入研究了重复数据消除所有阶段的最新方法，包括重复数据消除的新兴领域，如增量压缩、恢复、垃圾收集、安全性、可靠性等。

这项研究的主要贡献有三个方面。

- 我们研究了重复数据消除的背景和方法。通过研究重复数据消除和传统压缩方法之间的主要区别，我们展示了重复数据消除的主要功能和优势。
- 我们研究并分类了重复数据消除工作流每个阶段的技术水平，包括分块方法、分块的计算加速、指纹索引、增量压缩、数据恢复、垃圾收集、安全性和可靠性。基于对现有方法基于阶段的深入研究，我们对最新的重复数据消除技术进行了详细分类，这为基于重复数据消除的存储系统的重要设计问题提供了有用的见解。
- 我们讨论了重复数据删除的主要应用和行业趋势；为重复数据消除研究社区提供公开的开源项目、数据集和跟踪的集合；并概述了重复数据消除研究面临的开放问题和研究挑战。

值得注意的是，本研究中讨论的重复数据消除不同于数据库、数据挖掘和知识工程领域中的术语“重复数据消除”。在这些字段中，重复检测通常用于识别表示相同属性的记录 [33, 34]。这项研究的重点是消除相同的数据，从而节省存储系统中的存储空间。论文的结构如图所示 2。

部分 2 介绍了重复数据消除的背景和方法。第 3 节介绍了最先进的数据重复方法，并将其分为六类。第 4 节介绍了重复数据消除工作流程的关键阶段。部分 5 介绍了重复数据消除的主要应用，讨论了重复数据消除的行业趋势和未来研究方向，并介绍了一组公开可用的资源。

让研究团体研究重复数据删除技术。部分 6 总结全文。最后，在附录中，我们列出并解释了一些常见的

讨论首字母缩略词和术语。

## 2 冗余数据减少和重复数据消除

在本节中，我们首先通过介绍一个分类法来提供冗余数据减少的必要背景，该分类法显示了传统的数据压缩方法和重复数据消除方法，以及它们在过去几十年中的发展。然后，我们研究重复数据消除的关键特性，以展示它与传统压缩方法的不同之处及其原因。最后，我们回顾了典型重复数据消除系统的关键工作流程。

### 2.1 冗余数据约简入门

“数据压缩的目标是使用最少的位数尽可能准确地表示信息源（例如数据文件）。” [35, 36]. 一般来说，数据压缩可以分为两大类，无损和有损。无损压缩通过以可逆的方式识别和消除统计冗余来减少数据，如 GZIP [37] 和 LZW [19]. 有损压缩通过识别不必要的信息并不可挽回地删除它来减少数据，典型的是 JPEG 图像压缩 [38]. 本文重点介绍无损压缩类别，包括重复数据消除，因为通用存储系统需要无损压缩。

数据压缩理论是由克劳德·e·香农提出的，他在 1948 年的开创性论文《通信的数学理论》中引入了信息熵理论（简称“熵”），以定义信息出现概率（或不确定性）和数据冗余之间的关系 [39]. 这个熵理论建立了无损数据压缩的基本极限。具体地，具有可能值  $x_1$  的离散随机变量  $X$  的熵  $H, \dots, x_n$  和概率质量函数  $P(X)$  由香农定义为：

$$h(X) = E(I(X)) = \sum_{i=1}^n P(X_i) I(X_i) = -\sum_{i=1}^n P(X_i) \log_2 P(X_i) \quad (1)$$

这里  $E$  是期望值算子， $I$  是  $X$  的信息含量， $I(X)$  本身是随机变量，

而  $b$  是所用对数的基数（例如， $b=2$  指的是二进制表示）。例如，给定字符串 “abaaacabba”，字符 “a”、“b” 和 “c” 的出现次数分别为 6、3 和 1。因此，每个字符和字符串的熵计算如下：

$$\begin{aligned} h(a) &= \log_2(0.6) = 0.737 \text{ 位} \\ h(b) &= \log_2(0.3) = 1.737 \text{ 位} \end{aligned} \quad (2)$$

$$h(c) = \log_2(0.1) = 3.322 \text{ 位}$$

$$H(\{a, b, c\}) = H(a) \times 0.6 + H(b) \times 0.3 + H(c) \times 0.1 = 1.295 \text{ 位}$$

ASCII 需要 80 位来表示这 10 个字符

演员。方程式 (2) 意味着每个字符可以最大限度地压缩成 1.295 位，因此理论上整个字符串可以压缩成 12.95 位，尽管由于部分位是不可能的，所以需要更多。例如，我们可以用 “0”、“10” 和 “11” 分别编码 “a”、“b” 和 “c”；这将需要 14 位 (“01000011010100")。因此，香农熵理论揭示了信息消息可以用更少的比特来表示，这是冗余数据约简的本质。

早期的数据压缩方法使用基于统计模型的编码，也称为熵编码，在字节级识别冗余。最广泛使用的熵编码算法是哈夫曼编码 [40]，它使用频率排序的二叉树来生成用于熵编码的最佳前缀码。为了达到数据压缩率的基本极限（例如，前述的熵值），算术编码 [41]，最早由埃利亚斯在 20 世纪 60 年代提出 [42]，将整个消息编码成几个小数位，以获得更高的压缩率。霍夫曼编码将输入分离成分量符号，并用较短的代码替换每个分量符号。

随着世界上数字数据的不断增长，一种叫做基于字典模型的编码的新方法在 20 世纪 70 年代由伦佩尔和齐夫提出，由 LZ77 代表 [16] 和 LZ78 [17] 算法。这些通过在字符串级别识别冗余来简化和加速数据压缩：它使用滑动窗口识别重复的字符串，并用匹配字符串的位置和长度替换这些重复的字符串。20 世纪 80 年代后期，提出了 LZ 压缩方法的变体来提高压缩比（例如 DEFLATE 和 LZMA [20, 43, 44]）或加快压缩过程（例如，LZO [18] 和 LZW [19]）。

一般来说，熵编码方法需要在将频繁出现的字节编码成较短的比特之前对所有信息进行计数，这是不可伸缩的。同时，字典编码方法需要搜索所有字符串，以支持匹配和消除重复字符串。因此，熵编码和字典编码方法经常限制压缩

sion 窗口来权衡压缩比

LZ77 和霍夫曼编码) 使用 64KB 的压缩窗口大小，而最大窗口为

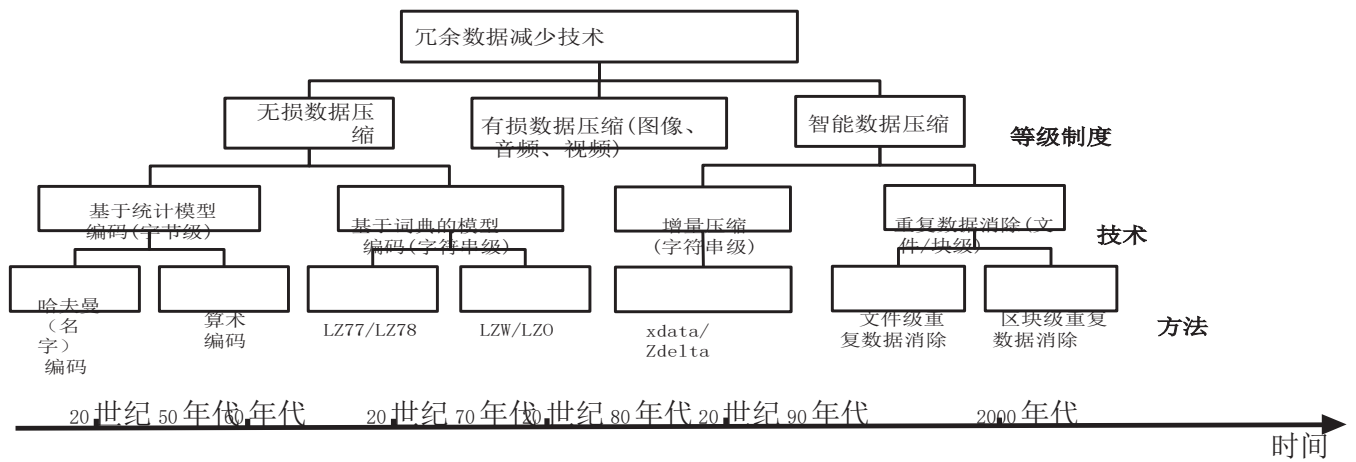


图3. 冗余数据减少技术的分类和每种方法的初步研究的大致日期。

bzip2 [21, 45]和7z [44]分别为900KB和1GB。尽管有一些通过预处理数据来增加压缩的技术，但大范围的压缩窗口大小会导致整体可压缩性的巨大变化[21, 46]。

Delta压缩是在20世纪90年代提出的，旨在压缩非常相似的文件（即文件的不同版本）或相似的块。它已被广泛应用于许多应用，如源代码版本控制[47-49]，远程同步[8, 50-52]和备份存储系统[53-55]。服用Xdelta [49]例如，对于类似的文件（或区块）A和B，xdata使用COPY/INSERT指令将匹配/不匹配的字符串记录到delta文件中，B。xdata可以通过使用文件B解码delta文件，A，B来快速恢复文件A。值得注意的是，delta压缩比重复数据消除早了十年或更长时间，但是最近，在重复数据消除中添加delta压缩越来越受到关注，这将在后面的章节中讨论3.4。

重复数据消除是在2000年代提出的，以支持大规模存储系统中更粗粒度的全局压缩（例如，文件[10]或8KB的数据块[11]）优于传统的压缩方法。它计算文件或块的加密安全的基于散列的指纹，然后通过匹配它们的指纹来识别副本。

数字3介绍了这些冗余数据减少技术的分类及其在过去几十年的发展。每种方法的日期都很接近，并表明研究开始的时间。在许多情况下，对每种方法的研究目前仍在继续。这些冗余数据减少技术是为了应对用户数字数据不断增长的规模而开发的，从熵编码发展到数据编码，再到增量压缩，现在又发展到重复数据删除。通过以越来越粗的粒度识别冗余，从字节级到字符串级，到块级，再到文件级，这些方法变得越来越可扩展。

## 2.2 重复数据消除的主要功能

在本小节中，我们将研究重复数据消除的关键功能。目前，使用最广泛的数据

重复数据消除方法通过计算然后匹配数据块的指纹来消除大规模存储系统中块级别的冗余[5, 7]。值得注意的是，文件级重复数据消除早些时候被提出[10, 56]，但该技术随后被块级重复数据消除所掩盖，因为后者具有更好的压缩性能[5, 12, 13, 57]。在某些情况下，文件级重复数据消除获得了更细粒度比较的大部分好处[5]。

低带宽网络文件系统（LBFS）[14]和Venti [11]是数据重复的两项开创性研究。LBFS被提议检测和消除大约8KB大小的重复组块（使用可变大小组块方法，详见第3.1节）并因此避免在低带宽网络环境中传输它们，而Venti则被提议移除重复的块并节省存储空间。具体来说，LBFS使用内容定义的组块来识别组块级别的重复，而Venti使用固定大小的组块。另一方面，Venti和LBFS都采用SHA1散列函数来计算组块的指纹，并且认为具有相同SHA1值的两个不同组块的概率远低于硬件比特错误的概率[11, 13, 14]。在过去的十年中，组块级重复消除和基于SHA1的指纹识别方案已经被学术界和工业界广泛接受和采用[13, 58-64]。已经提出了碰撞可能性的问题[65]和反驳[66]，如下文进一步讨论的。

块级重复标识。虽然近年来数字数据的规模一直在增长，但信息熵并没有成比例地增长。例如，大量数据可能是重复复制某些数据并将其存储在备份/归档存储系统中的结果[7]。此外，传统的压缩方法使用字节级滑动窗口在相对较小的窗口内找到用于重复识别的匹配字符串；通过比较，重复数据消除将输入分成存储系统中不重叠且独立的区块。

一般来说，数据的分块方法



重复数据消除包括固定大小分块和可变大小分块。固定大小分块只是根据内容的偏移量(即位置)将输入分成固定大小的组块。这种方法很简单,但是由于已知的边界移位问题,可能无法识别大量冗余,其中数据流中的微小变化(例如,插入/删除一些字节)可能导致所有块的边界移位。由于数据块包含略有不同的内容,因此无法进行重复数据消除[14, 60, 67, 68]。可变大小分块,也称为内容定义分块,根据内容本身将输入分成可变大小的块;这在很大程度上解决了边界移动问题,也是目前使用最广泛的分块方法。组块将在第3节中研究3.1。

加密安全的基于哈希的指纹识别,也称为哈希比较[65, 66]。指纹技术简化了重复识别的过程。在一些早期的数据简化方法中(例如LZ压缩[16, 17]和Xdelta[49]),首先通过计算出的弱散列摘要来匹配这两个副本,然后通过逐字节比较来进一步确认。在重复数据删除系统中,重复数据完全由其基于加密哈希的指纹(例如,SHA1、SHA256)来表示,匹配的指纹意味着它们所表示的内容很可能彼此相同。这里指纹指的是一系列加密散列函数[69]这具有实际上不可行的关键属性,即(1)找到两个具有相同散列的不同消息,以及(2)从给定散列生成消息。

根据“生日悖论”[11, 70],给定SHA1对的碰撞概率可以计算如下:

哈希计算:CA = 哈希(内容) (CA长度为m位)

$$\text{哈希冲突: } p \leq \frac{n(n-1)}{2 \times 2^m} \quad (n \text{ 是块的数量}) \quad (3)$$

其中CA是使用“哈希”函数的内容的m位指纹。桌子1显示了根据等式(3)具有不同数量的唯一数据。基于8KB的平均区块大小和SHA1指纹,在电子商务规模的存储系统中执行重复数据消除时,哈希冲突的概率小于10-20(约2-67)。相比之下,在计算机系统中,硬盘驱动器出错的概率约为10<sup>-12</sup>到10<sup>-15</sup>[71, 72],这远远高于前面提到的重复数据消除中SHA1指纹冲突的概率。因此,SHA1已成为重复数据消除最广泛使用的指纹算法,因为大多数现有方法,如LBFS[14],Veti[11]和DDFS[13],请考虑哈希冲突概率足够小,以便在PB级存储系统中应用重复数据消除时可以忽略。然而,亨森[65]然而,指出基于散列的比较方法没有风险

表 1

在唯一数据大小不同且平均区块大小为8KB的存储系统中,对SHA-1、SHA-256和SHA-512进行哈希冲突概率分析。

| 的大小<br>唯一数据 | SHA-1<br>160 位 | SHA-256<br>256 位 | SHA-512<br>512 位 |
|-------------|----------------|------------------|------------------|
| 1GB (230B)  | 10-38          | 10-67            | 10-144           |
| 1TB (240B)  | 10-32          | 10-61            | 10-138           |
| 1PB (250B)  | 10-26          | 10-55            | 10-132           |
| 1EB (260B)  | 10-20          | 10-49            | 10-126           |
| 1ZB (270B)  | 10-14          | 10-43            | 10-120           |
| 1YB (280B)  | 10-8           | 10-37            | 10-114           |

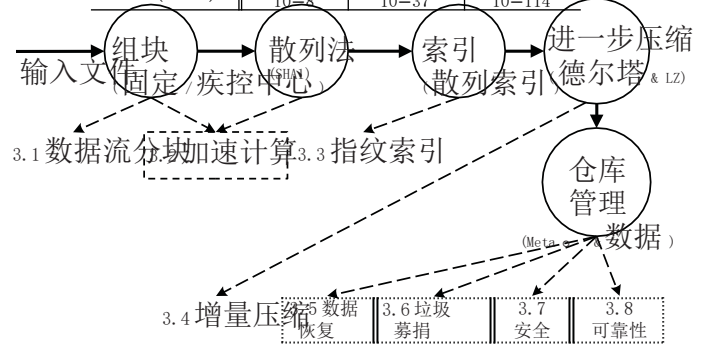


图 4. 重复数据删除的基本工作流程。

免费的。Black 等人[66]通过解决 Henson 提出的问题,提供了几个支持哈希比较的论点。最近,在一些重复数据消除系统(如ZFS)中,更强的哈希算法(如SHA256)已被考虑用于指纹识别[73]和Dropbox[74],进一步降低哈希冲突的风险。内容定义的分块和基于安全散列的指纹在商业存储系统中被广泛使用[7, 13, 58, 59, 62, 64]并被称为重复数据消除的关键功能。桌子2显示了Microsoft对大规模真实工作负载(即TB-PB规模的数据集)的数据缩减率进行的分析摘要[5, 6],EMC[7, 8]和密茨大学[70]。这里

区块级重复数据消除基于内容定义的

组块[14]而增量压缩是基于Xdelta编码的[49]。表2表明,块级重复数据消除检测到的冗余比文件级多得多,并且重复块在计算机系统中非常丰富,数据缩减率约为二级存储、主存储和高性能计算数据中心分别为99.97%、42.68%和20.30%。

## 2.3 重复数据删除的基本工作流程

典型的重复数据消除系统遵循分块、指纹识别、索引、进一步压缩和存储管理的工作流程,如图所示4。重复数据消除系统中的存储管理可以进一步分为几类,包括数据恢复(碎片消除)、垃圾收集、可靠性、安全性等。因为数据重复数据消除旨在实现大规模存储系统中的全局冗余数据减少,如第3节所述2.1,有很多有趣的研究

表 2  
工业界和学术界最近发表的关于大规模真实数据集冗余数据减少率的分析摘要。

| 学院    | 工作量                  | 大小      | 冗余数据减少方案             | 减速比    |
|-------|----------------------|---------|----------------------|--------|
| 微软    | 857 台台式计算机[5]        | 162 TB  | 文件级重复数据消除内部用户        | 大约 21% |
|       |                      |         | 区块级重复数据消除内部用户 (8KB)  | 大约 42% |
|       |                      |         | 跨用户的文件级重复数据消除        | 大约 50% |
|       |                      |         | 跨用户的区块级重复数据消除 (8KB)  | 大约 68% |
|       | 15 台微软文件服务器[6]       | 6.8 TB  | 文件级重复数据消除            | 0~16%  |
| 电磁兼容性 | 超过 1 万件产品<br>备份系统[7] | 700 TB  | 区块级重复数据消除 (8KB)      | 69~93% |
|       |                      |         | 区块级重复数据消除 (8KB)      | 85~97% |
|       | 6 个大型备份数据集[8]        | 33 TB   | 重复数据消除+增量压缩          | 66~82% |
|       |                      |         | 重复数据消除+增量压缩+ GZIP 压缩 | 74~87% |
|       |                      |         | 文件级重复数据消除            | 5~10%  |
| 美因茨大学 | 四个高性能计算数据中心[70]      | 1212 TB | 区块级重复数据消除 (8KB)      | 20~30% |
|       |                      |         | 区块级重复数据消除 (8KB)      | 20~30% |

与重复数据消除工作流的每个阶段相关的问题和解决方案，这些问题和解决方案可以分类如下，并在本文的剩余部分进行详细阐述。

- 数据流分块: 如何设计一种高效的分块算法来最大限度地检测数据流中的冗余，以实现重复数据删除，这是一个重要的研究和实际问题，本节将深入研究这一问题及其解决方案 3.1。
- 加速计算任务: 重复数据消除系统中的分块和指纹识别占用大量计算资源。减少这种开销是一个新出现的具有挑战性的问题，将在第 3 节中讨论 3.2。
- 指纹索引。随着用户数据的大小不断从 TB 增长到 PB 甚至 EB，代表用户数据的指纹的总大小将很快超过主内存。这就产生了如何有效地存储和索引存储在磁盘上的用户数据块的这些指纹的问题，这是一个越来越重要和具有挑战性的研究和实际问题。我们将在第 3 节全面研究这个问题的现有解决方案 3.3。
- 进一步压缩。如何消除重复数据删除系统中非重复但非常相似的数据块之间的冗余，即增量压缩，是另一个有趣而重要的研究问题。如图所示 4，增量压缩和传统压缩（例如 LZ 压缩）方法属于进一步压缩范畴。LZ 压缩直观且易于实现 [7, 8, 13] 尽管增量压缩是一种可选方法，因为它给重复数据消除系统带来了新的挑战，这将在第 3 节中进行深入研究 3.4。
- 数据恢复: 在基于重复数据消除的存储系统中，文件或数据流的区块在重复数据消除后可能会在物理上分散存储，这可能会显著降低恢复（读取）的性能。将全面介绍提高基于重复数据消除的存储系统的读取性能的最先进的工作

在第 3 节中学习 3.5。

- 垃圾收集: 在经过重复数据消除的系统中，一些数据区块将由许多文件共享，因此，确定在删除文件时哪些区块可以回收是一个新的挑战。垃圾收集问题和现有解决方案将在第 3 节中进行研究 3.6。
- 保安。不同用户的数据之间的重复数据消除可能会导致潜在的安全风险，其中一个用户的隐私和敏感信息会泄露给另一个用户，反之亦然。第 3 节将深入研究安全重复数据消除问题和初步解决方案 3.7。
- 可靠性: 重复数据消除降低了存储系统的可靠性，因为丢失几个关键数据块会导致许多引用的文件/备份丢失。第 3 节将全面研究针对重复数据消除系统的可靠性增强解决方案 3.8。

### 3 重复数据删除: 关键技术的深入探索

在本节中，我们将深入研究有关重复数据消除的最新作品，以了解它们的关键和显著特征。这些功能将现有的重复数据删除技术分为八类（子部分 3.13.8）基于重复数据消除过程的工作流程，如图所示 4。每一个分支都将全面研究和讨论每一个类别中的现有方法。

#### 3.1 数据流的分块

分块是重复数据消除过程的第一步，在这一过程中，文件或数据流被分成小块数据，以便对每个数据进行指纹识别（见图 4）。最简单的分块方法是将文件/数据流切割成相等的固定大小的块，这种方法被称为固定大小分块 (FSC)。在 FSC，如果文件或数据流的一部分（无论多小）被插入或删除操作修改，不仅包含

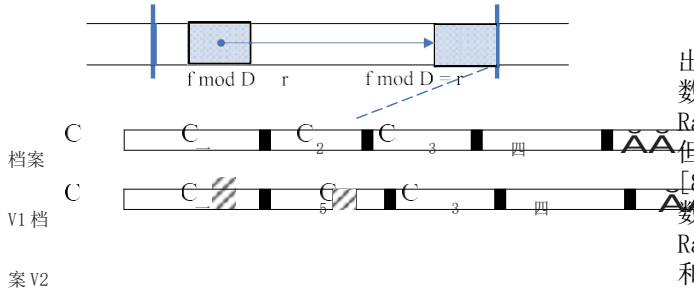


图5. CDC算法的滑动窗口技术。滑动窗口的散列值  $f$  是通过拉宾算法计算的。如果哈希值的最低  $\log_2 D$  位匹配预定义值  $r$ , 即  $f \bmod D = r$ , 则偏移量被标记为块断点(也称为断点)。这里, 文件 V2 的块 C5 内的阴影区域表示文件 V2 中相对于文件 V1 新插入的内容。

修改的部分改变了, 但是所有后续的数据块也会改变, 因为所有这些块的边界都被移动了。这可能导致原本相同的数据块(在修改之前)完全不同, 从而导致基于 FSC 的重复数据消除的重复识别率显著降低。

为了解决这个边界移动问题[12, 14, 75, 76]在 LBFS 提出了内容定义分块(CDC)算法, 也称为基于内容的断点分块[14], 对文件或数据流进行分块以进行重复识别。具体来说, CDC 对文件内容使用滑动窗口技术, 并计算哈希值(例如, 拉宾指纹[77])的窗口, 如图5。如果该滑动窗口的哈希值满足某个预定义的条件, 则确定块断点。因此, 对于在文件 V1 的组块 C2 上被修改的文件 V2 的组块, CDC 算法仍然可以识别其内容未被修改的组块 C3 和 C4 的正确边界。目前, 拉宾算法[70, 77, 87]是 CDC 中计算滑动窗口哈希值最广泛使用的算法, 用于重复数据删除。具体地, 滑动窗口的拉宾签名(指纹)(字节序列  $B_1, B_2, \dots, B_\alpha$ )

流, 定义为:

$$\text{拉宾}(B_1, B_2, \dots, B_\alpha) = \sum_{x=1}^{\alpha} B_x p^{\alpha-x} \bmod D \quad (4)$$

其中  $D$  是平均块大小,  $\alpha$  是滑动窗口中的字节数。拉宾签名是通过滚动散列算法获得的, 因为它能够以增量方式计算签名。对于滑动窗口中的子字符串, 签名可以从前一个值开始递增计算, 如下所示:

$$\begin{aligned} \text{拉宾}(B_{i+1}, B_{i+2}, \dots, B_{i+\alpha}) &= \left( \sum_{x=i+1}^{i+\alpha} B_x p^{\alpha-x} \bmod D \right) \\ &= \left( \left[ \sum_{x=i}^{i+\alpha-1} B_x p^{\alpha-x-1} \right] p + B_{i+\alpha} \right) \bmod D \\ &= \left( \left[ \text{拉宾}(B_i, B_{i+1}, \dots, B_{i+\alpha-1}) p + B_{i+\alpha} \right] \right) \bmod D \end{aligned} \quad (5)$$

因此, 构建新的 Rabin 散列可以从旧的散列中快速计算出来, 只需要 2 次 XOR、1 次 OR、2 次左移和每个字节 2 次数组查找的操作[68]。在重复数据删除系统中, 虽然 Rabin 在滚动散列属性方面满足了内容定义分块的需求, 但其效率可以提高。例如, 最近的一项名为 QuickSync [88], 表明基于 Rabin 的 CDC 对于移动云存储中基于重复数据消除的同步来说计算成本相当高。一般来说, 基于 Rabin 的 CDC 存在三个主要缺陷: 大块大小方差、计算开销和重复检测不准确。

桌子 3 总结了基于 CDC 的重复数据消除的最新方法, 这些方法解决了基于 Rabin 的 CDC 的这三个不足, 下面将详细介绍。

通过对 CDC 的最大/最小块大小进行限制来减少块大小差异。在平均块大小为 8KB 的基于 Rabin 的分块方法中, 块大小  $X$  的累积分布遵循以下指数分布[68]:

$$p(X \leq x) = F(x) = 1 - e^{-x/8192}, \quad x \geq 0 \quad (6)$$

方程式 (6) 表示将有许多非常小或非常大的块(例如,  $< 1\text{KB}$  或  $> 64\text{KB}$ )。较小大小的数据块会导致重复数据消除的元数据开销较高, 因为此开销与数据块数量成正比; 而较大大小的数据块会降低重复数据消除率[75]因为大块往往会隐藏重复项而不被检测到。因此, LBFS 对内容定义的组块的最小和最大组块大小(例如, 2KB 和 64KB)施加了限制[14]。然而, 这种解决方案可能会在分块过程中产生不再由内容定义的强制切割点。

“两个门槛, 两个约数”(TTTD) [78] 引入了一个额外的分块阈值, 它具有更高的找到切割点的概率, 并减少了块大小的差异。回归组块[6]与 TTTD 类似, 但它使用多个阈值来减少最大块大小时的强制断点声明。MAXP [80, 89, 90] 将固定大小区域中的极值视为切割点, 这也会导致较小的块大小差异。指纹差异[79, 91]使用较小的预期块大小来检测更多的重复, 然后合并连续重复或唯一的块, 以分摊由较小的预期块大小产生的额外元数据开销。

**减少计算以加速分块过程。**由于滑动窗口上拉宾指纹的频繁计算是耗时的,

已经提出了 Rabin 的许多替代方案来加速 CDC 的重复数据消除过程[68, 81, 82]。SampleByte [81] 旨在提供快速分块

用于细粒度的网络冗余消除。它使用

一个字节声明用于分块的指纹(而

表 3  
针对重复数据消除的内容定义分块的最先进方法。

| 名字                                     | 来源              | 独特的功能                      | 分类学             |
|--|-----------------|----------------------------|-----------------|
| LBFS                                   | SOSP'01 [14]    | 最大/最小/平均区块大小为 64KB/2KB/8KB | 最大/最小块大小的限制     |
| TTTD                                   | 惠普技术。报告 [78]    | 最大和最小块大小的阈值                |                 |
| 回归分块                                   | USENIX'12 [16]  | 最大块大小的多个阈值                 |                 |
| 指纹差异                                   | MSST'06 [79]    | 将连续的重复块或唯一块合并成更大的块         |                 |
| MAXP                                   | 微软技术。报告 [80]    | 将固定大小区域中的对称极值视为切点          |                 |
| 样本字节                                   | NSDI'10 [81]    | 跳过预期块大小的一半，以加速分块           | 提高疾控中心的速<br>度   |
| 齿轮                                     | 业绩'14 [68]      | 通过随机整数表减少散列计算              |                 |
| 自动曝光装置                                 | Infocom'15 [82] | 使用非对称滑动窗口识别切割点             |                 |
| 飞跃式疾控中心                                | MSST'15 [83]    | 使滑动窗口向前跳跃以减少计算             |                 |
| 双峰组块                                   | FAST'09 [75]    | 重新分块唯一但重复的相邻块              | 进一步重新分块非重<br>复块 |
| 子块                                     | SYSTOR'11 [84]  | 将所有独特的块重新分成更小的块            |                 |
| 流化床燃烧<br>(Fluidized-Bed<br>Combustion) | 吉祥物'10 [85]     | 以高频率对非重复块进行重新分块            |                 |
| 磁铃力学<br>(magnetohydrodynami<br>cs)     | ICPP'13 [86]    | 动态合并和重新分块非重复的块             |                 |

Rabin 使用滑动窗口)并将最小尺寸从预期平均值四分之一增加到一半，与 LBFS 相比 [14] 在搜索每个块的切点之前跳过最小大小。齿轮 [68] 通过使用随机整数表来映射内容的 ASCII 值，使用较少的操作来生成滚动哈希，从而实现更高的分块吞吐量，同时获得与 Rabin 相当的分块精度。不对称极值 [82] 使用非对称滑动窗口，而不是像 MAXP 中那样使用对称滑动窗口 [80]，以将极值识别为切点，这进一步降低了分块的计算开销，从而实现了高分块吞吐量，同时保持了低块大小变化。于等 [83] 调整了用于选择组块边界的功能，使得如果不满足弱条件，滑动窗口可以向前跳转，节省了不必要的计算步骤。

通过重新分块非重复块来提高重复检测的准确性。如图所示 5，CDC 方法简单地确定块边界元，如果散列 (例如，Rabin [77]) 匹配预定义的值，这有助于识别文件 V1 和 V2 中重复的区块 C3 和 C4，但无法识别区块 C2 和 C5 之间的冗余。更具体地，CDC 方法不能准确地找到改变的区域 (例如，文件 V2 的块 C5 中的阴影区域) 和两个相似文件 (例如，文件 V1 和 V2) 之间的重复区域 (例如，块 C5 中的空白无阴影区域) 之间的边界。因此，已经提出了许多重新组块方法来进一步划分非重复组块 (例如，图中的组块 C2 和 C5) 到更小的区域以检测更多的冗余 [75, 84–86]。

双峰组块 [75] 首先将数据流分成大块，然后将非重复但重复相邻的块重新分成更小的块，以检测更多的冗余。子块 [84] 类似于双模分块，但它会将所有非重复数据块重新分块，以获得更高的重复数据消除率。基于频率的组块 (FBC) [85] 使用统计块频率估计算法来识别频繁的块，然后将这些块重新分成更小的块。

检测更多重复项。元数据利用重复数据消除 (MHD) [86] 也类似于双峰分块，但它通过动态地将多个非重复块合并成一个大块，同时将非重复但重复相邻的块划分成较小的块，而双峰只做后者，进一步降低了重复数据删除的元数据开销。

MAXP [80, 89, 90] 和 SampleByte [81] 是两种基于非滚动散列的分块算法，它们被设计用于以精细粒度 (例如 64 或 128 字节) 消除冗余网络流量。MAXP 是为了解决 Rabin 中由于强制最小块大小而导致的去重复率降低的问题而提出的 [80]。它用于网络重复数据消除，因为它的区块大小差异较小 [89]。SampleByte 旨在加速内容定义的分块，以提高资源受限设备 (如移动智能手机) 中冗余消除的能效 [81]。

总之，强制最小和最大数据块大小的 CDC 方法被广泛接受，并因此用于基于重复数据消除的存储系统 [5, 7, 13, 92]。对最大/最小数据块大小施加更多限制虽然易于实施，但只会略微提高重复数据消除率。重新分块方案显著提高了重复数据消除率，但代价是耗时的重新分块过程以及对非重复块及其重新分块的子块的额外存储管理。其他快速分块方法，如 Gear [68]，AE [82] 和 SampleByte [81] 如果用户需要更高的重复数据消除吞吐量，可能会推荐使用。

穿插元数据的影响。当元数据散布在数据中时，未改变数据的分块会导致不同的块 [98]。元数据可以采取不同的形式。一种类型发生在存在固定大小块的头时 (例如，对于虚拟磁带)，这导致元数据相对于内容定义的块移动；这些可以由重复数据消除系统检测到，并在重复数据消除之前剥离。另一种类型是 tar 聚合文件或备份文件中包含的文件之前的每文件元数据。当每个文件的元数据包含时间戳、序列号，



表 4  
加速重复数据消除计算任务的最先进方法。

| 名字       | 来源              | 独特的功能                           | 分类学     |
|----------|-----------------|---------------------------------|---------|
| THCAS    | ICPADS' 09 [93] | 重复数据消除的流水线 CPU 限制、I/O 限制、网络通信任务 | 基于多核的   |
| HPDS     | USENIX' 11 [62] | 基于 FSC 的重复数据消除任务的事件驱动客户端管道      |         |
| 马等。      | MSST' 12 [94]   | 重复数据删除、压缩、加密等自适应管道。             |         |
| 重复数据消除   | NAS' 12 [76]    | 流水线化重复数据消除任务，并进一步并行化 CDC 和指纹识别  |         |
| StoreGPU | HPDC 08[95]     | 开发具有共享内存管理的重复数据消除 GPU 库         | 基于图形处理器 |
| 碎纸机      | FAST' 12 [96]   | 开发一个流水线和内存合并的 GPU 加速框架          |         |
| 幽灵       | PMAM' 12 [97]   | 将重复数据消除处理卸载到主存储中的 GPU           |         |

或其他可变内容，周围的区块将不会进行重复数据消除。这种元数据可以通过对输入进行预处理来解决(林等人将其称为迁移性 tar98])或创建便于重复数据消除的文件格式，将元数据与数据分开。

### 3.2 计算任务的加速

如图所示 5 和科 3.1 数据重复是一个计算密集型过程，包含两个耗时的计算任务，即内容定义的分块和基于安全哈希的指纹。前者由疾控中心将数据流分成几个块，后者为每个块计算一个加密摘要(即指纹)，以唯一地表示它用于重复检测。因此，重复数据删除过程的分块和查找打印阶段需要计算哈希(例如 Rabin 和 SHA1)，这可能会延长基于重复数据删除的存储系统中的写入延迟[95, 96, 99]，尤其是在使用基于闪存的设备的高性能主存储系统中，以增加内存中的处理能力[97]。

虽然重复数据删除的哈希计算耗时且占用大量 CPU，但基于多核/众核处理器或 GPU(图形处理单元上的通用计算)处理器的现代计算机系统正在提供越来越多的计算资源[95]。同时，重复数据消除过程可以分为几个独立的子任务，如图所示 5。利用这些子任务之间的并行性[96, 100]可以通过充分利用现代计算机系统中丰富的计算资源来加速重复数据删除。请注意，第节中讨论的方法 3.1 通过改进内部分块过程(例如，使用新的哈希算法)来提高 CDC 速度，而本小节主要讨论通过利用多核处理器或 GPU 设备来利用 CDC 过程的并行性，从外部加速基于 Rabin 的 CDC 的方法。

我们的研究表明，加速计算密集型重复数据消除任务的最先进方法主要基于利用多线程重复数据消除的并行性[62, 100]或将重复数据消除过程集成到 GPU 硬件架构中[95, 96, 101]。这些总结在表中 4 并在下面详细说明。

- 多线程方法。THCAS [93]在其重复数据消除系统中提出了一个 CPU 限制(即分块和指纹识别)、I/O 限制(即写入)和网络通信任务的存储管道。重复数据消除[76]类似于 THCAS，但它进一步并行化了分块和指纹识别的子任务，从而实现了更高的吞吐量。郭等[62]提出一个事件驱动、多线程的客户端-服务器交互模型，以管道化基于 FSC 的重复数据消除。马等[94]针对基于 FSC 的重复数据消除系统中的指纹识别、压缩和加密等计算子任务，提出一种自适应流水线模型。
- 基于 GPU 的方法。在许多计算密集型应用中，特别是在高性能存储系统中的哈希和密码计算应用中，图形处理设备表现出比中央处理器更强的计算能力。StoreGPU [95, 101]和碎纸机[96]充分利用 GPU 的计算能力，在重复数据删除中加速流行的计算密集型原语(即分块和指纹)。Similarly, GHOST [97]将分块、指纹和索引等重复数据消除任务卸载到 GPU，以消除高性能主存储系统中的计算瓶颈。

总之，通过流水线化重复数据消除任务以及并行化分块和指纹识别，基于多线程的解决方案可以很容易地在具有多核处理器的计算机系统中实现[93, 100]。基于 GPU 的解决方案可以提供更高的吞吐量，但需要额外的硬件成本[95, 96, 101]。另一种方法不影响计算任务的速度，但允许备份系统扩展到其他客户端，这种方法是将分块卸载到客户端；这将在第节的行业趋势中讨论 4.2。

### 3.3 指纹索引

在对数据流进行分块和指纹识别后，将对块指纹进行索引，以帮助确定重复和非重复数据块，这是重复数据消除过程的一个关键阶段。早期的重复数据消除系统将整个块指纹索引存储在内存中，以便快速识别重复数据[14]。

随着数据量、指纹总数以及索引大小的爆炸式增长

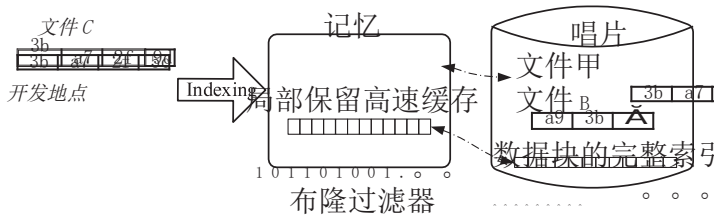


图 6. 典型的基于位置的指纹索引方法:DDFS [13].

呈指数级增长, 快速溢出重复数据消除系统的 RAM 容量。这可能导致频繁访问低速磁盘进行指纹索引查找, 从而严重限制重复数据消除系统的吞吐量。例如, 要备份 1PB 的唯一数据集, 并假设平均区块大小为 8KB, 将生成大约 2.5TB 的 SHA-1 指纹(每个区块 160 位)。2.5TB 指纹加上每个数据块的额外位置和索引消息(例如 8 字节), 将太大, 无法完全存储在典型重复数据消除系统的主内存中。由于随机存取磁盘索引比随机存取内存慢得多, 频繁存取磁盘指纹将导致系统吞吐量变得低得不可接受。例如, 一些重复数据消除系统[11, 13, 58]报告磁盘指纹索引的访问吞吐量约为 1-6mb/秒, 这成为这些系统中的严重性能瓶颈。因此, 高效的指纹索引方案对于大规模重复数据消除系统是必要的。根据所使用的具体方法, 指纹索引方案可以实现精确重复数据消除或近似重复数据消除[102, 103]。前者意味着消除了所有重复块, 而后者则以略微降低的重复检测精度(即, 少量重复块未被检测到)换取更高的索引查找性能和更低的内存占用。目前, 有四大类方法可加快重复数据消除的索引查找过程并缓解磁盘瓶颈, 即基于位置、基于相似性、闪存辅助和集群的重复数据消除方法。

桌子 5 介绍了这四类用于重复数据消除系统的最新指纹索引方法, 并对其进行了详细阐述下一个。

**基于位置的方法。**重复数据消除上下文中的局部性是指观察到备份流中相似或相同的文件, 例如 A、B 和 C(因此它们的数据块), 在多个完整备份中以几乎相同的顺序出现的概率非常高[13, 75]。挖掘这一位置进行重复数据消除索引可以提高内存利用率, 减少对磁盘索引的访问, 从而缓解磁盘瓶颈。

数字 6 显示了一个基于位置的方法的例子, DDFS [13]。这种众所周知的重复数据消除系统通过按照备份流的顺序存储块(例如, 块', 充分利用了这种局部性

文件 A 的指纹 3b、a7、2f、5c) 在磁盘上。在查找文件 C 的指纹“3b”时, DDFS 将预取指纹 3b、a7、2f、5c, 并在 RAM 中保留这一局部性, 这有助于减少以后查找“a7”和“2f”的指纹时对磁盘索引的访问。通常, DDFS 将非重复块存储在几个称为容器的大型固定大小存储单元中, 以保持备份流的局部性。DDFS 也使用布隆过滤器[22]以快速识别新的(即, 非重复的)组块, 避免对已知不存在的组块的索引查找; 这有助于补偿没有或很少局部性的情况。布隆过滤器[22]是一种节省空间的数据结构, 它使用带有几个独立散列函数的位数组来表示一组项目(例如指纹)的成员资格。

稀疏索引[58]通过对内存中的块指纹索引进行采样, 而不是使用 Bloom 过滤器, 来提高 DDFS 内存利用率[22]如 DDFS, 它将内存使用量减少到不到 DDFS 的一半。萨姆[105]首先结合全局文件级重复数据消除和本地块级重复数据消除, 然后利用大小、类型、位置等文件语义, 以优化指纹索引。MAD2 [104]使用布隆过滤器阵列作为重复数据消除的快速索引, 同时还保留了缓存中指纹的位置。HPDS [62]利用备份流的固有局部性和渐进采样索引方法, 进一步降低指纹索引的内存开销。DDFS [13]通过按照容器中存储的唯一块的顺序存储和预取来捕获局部性。与 DDFS 不同, 块位置缓存(BLC) [106]通过利用长期备份系统中最近备份的位置来提高索引性能。

**基于相似性的方法。**重复数据删除上下文中的相似性是指文件或数据流与先前相似文件或数据流的相似性。常见的相似性检测技术是用组块指纹集的最大值或最小值来表示文件[60]。因此, 所选指纹可用于构建主索引, 并最小化重复数据消除索引的内存开销, 尤其是对于局部性很小或没有局部性的数据集。极端宁滨[60]是一种基于相似性的方法, 通过利用文件相似性来实现对每个文件的单个磁盘索引访问以进行区块查找, 从而提高了重复数据消除的可扩展性。

极端宁滨证明, 根据布罗德定理, 文件 S1 和 S2 共享相同的代表性指纹的概率与其相似度密切相关[115]。也就是说, 考虑 S1 和 S2 的两个集合,  $H(S1)$  和  $H(S2)$  分别是 S1 和 S2 元素的散列的对应集合, 其中 H 是从最小独立排列族中一致地和随机地选择的。让最小(S)表示整数集合 S 的最小元素。然后:

表5  
用于重复数据消除的最先进的指纹索引方法。

| 名字           | 来源                 | 计划  | 独特的功能               | 分类学      |
|--------------|--------------------|-----|---------------------|----------|
| DDFS         | FAST' 08 [13]      | 确切的 | 布隆过滤器+局部保留缓存        | 基于位置的    |
| 稀疏索引         | FAST' 09 [58]      | 近似的 | 稀疏采样+局部性利用          |          |
| MAD2         | MSST' 10 [104]     | 确切的 | 布隆过滤器数组+利用局部性       |          |
| 地对空导弹        | ICPP' 10 [105]     | 确切的 | 利用大小、类型、位置等文件语义。    |          |
| HPDS         | USENIX' 11 [62]    | 近似的 | 渐进式抽样索引+利用局部性       |          |
| BLC          | SYSOR' 13 [106]    | 确切的 | 缓存最后一个备份流的位置        | 基于相似性    |
| 极端宁滨         | 吉祥物' 09 [60]       | 近似的 | 将文件代表性指纹用于主索引       |          |
| Aronovich 等人 | SYSOR' 09 [107]    | 近似的 | 构造卡普-拉宾相似签名         |          |
| 筒仓           | USENIX' 11 [63102] | 近似的 | 相似性和局部性的联合利用        |          |
| 重复数据消除 1     | MSST' 10 [108]     | 确切的 | 存储在固态硬盘上的区块指纹       | 闪光辅助的    |
| ChunkStash   | USENIX' 10 [61]    | 确切的 | 布谷鸟散列+存储在闪存上的指纹     |          |
| BloomStore   | MSST' 12 [109]     | 确切的 | 基于布隆过滤器的键值存储+闪存辅助   |          |
| HYDRAsTOR    | FAST' 09 [59]      | 确切的 | 基于指纹前缀的无状态路由        | 群集重复数据消除 |
| 防止重复数据删除     | IPDPS' 10 [110]    | 确切的 | 通过共享全局索引进行后处理重复数据消除 |          |
| 类似索引         | FAST' 11 [111]     | 近似的 | 利用相似性的超块路由          |          |
| -重复数据删除      | 中间件' 12 [112]      | 近似的 | 基于相似性和局部性的数据路由      |          |
| 凯泽等人         | MSST' 12 [113]     | 确切的 | 设计一个联合分布式索引         |          |
| 生产           | SOCC' 12 [114]     | 近似的 | 基于概率的数据路由           |          |

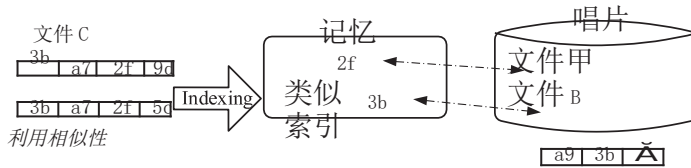


图7. 典型的基于相似性的指纹索引方法: 极限宁滨[60].

$$pr[\min(h(s)) = \min(h(s))] = s1 \cup s2$$

(7)

$$S1 \cup S2$$

数字7显示了Extreme宁滨如何利用文件相似性的示例，其中两组区块指纹3b、a7、2f、5c和3b、a7、2f、9d分别属于文件A和C。这里，文件相似性由文件的散列集中的最小指纹来表示，该文件的前缀位表示那里所有指纹的相同前缀位中的最小值。因此，在检测到文件C的最小指纹“2f”与文件A的指纹相同的情况下，我们可以认为这两个文件相似，然后检测文件A和C之间的重复块，这避免了对文件C的块指纹进行全局索引。

Aronovich等人[107]利用大规模重复数据消除系统中备份流的相似性。他们将数据流分成多个16MB的大数据块，为每个数据块构造一个签名来检测相似的数据块。对于每一对检测到的similar块，执行逐字节比较，以识别和消除重复数据。SiLo[102]通过首先利用数据段（一组组块）的相似性来减少RAM中主索引的空间，然后挖掘局部性来通过概率相似性检测来增强重复检测（参见等式(7)）。

最近的一篇论文，DeFrame[103]，提供了一个通用的开源框架，用于全面研究和评估上述许多问题

讨论了基于局部性和相似性的方法。它进一步澄清了局部性概念，将局部性分为两类，即逻辑局部性和物理局部性，这意味着分别是重复数据消除前后备份流的区块（指纹）序列。DDFS[13]HPDS[62]，以及ChunkStash[61]在稀疏索引时利用物理局部性[58]，SiLo[102]和BLC[106]ex-plot逻辑局部性。DeFrame还讨论了重复数据消除率、索引内存占用和恢复性能之间的设计权衡，为重复数据消除系统的设计决策提供了详细的指导。

闪光辅助方法。因为磁盘上指纹的查找吞吐量受限于昂贵的磁盘寻道操作（仅约100 IOPS），所以随机存取闪存被提议作为磁盘的替代品，以提供用于指纹索引的高吞吐量I/o（约100,000 IOPS）[116]。闪存指纹的内存高效和高性能主索引设计，称为键值存储[109,116,117]，进一步用于这些闪存辅助的重复数据消除索引方法。

重复数据消除1[108]和ChunkStash[61]将区块指纹存储在闪存而不是硬盘上，以加速索引查找过程。ChunkStash预先服务于内存中的备份流位置，以提高内存利用率并减少对闪存索引的访问。此外，布谷鸟哈希[118]被ChunkStash用来将指纹索引组织成内存中的键值存储，这比DDFS的Bloom过滤器更有效。更重要的是，ChunkStash使用了一个额外的“Stash”来避免布谷鸟散列表中索引指纹的可能循环，因此得名“stash”。FlashStore[116]和skippystash[117]集成布隆过滤器的数据结构[22]和哈希表，以获得更好的键值存储性能。BloomStore[109]通过采用基于布隆过滤器的索引，进一步分摊了内存开销并提高了查找/插入性能

闪存指纹的有效键值存储结构。

群集重复数据消除。上述大多数方法消除了单个节点上的重复，这限制了数据重复的吞吐量和可扩展性。这一限制导致了由多个节点组成的群集重复数据消除系统的发展[59, 111, 112]。群集重复数据消除的基本思想是通过数据路由方案将数据流从备份客户端分配到多个重复数据消除节点，该方案支持节点间负载均衡，并在单个节点中实现独立的节点内重复消除[112]。

HYDRAsTOR [59]以粗块粒度(64KB)执行重复数据消除，并使用分布式哈希表(DHT)基于指纹的前缀将块级别的数据分发到存储节点。DEBAR [110]首先在文件写入服务器磁盘时对其进行部分重复数据消除，然后通过节点间的全局指纹索引在区块级别完成后处理重复数据消除。请注意，极端宁滨[60]和MAD2 [104]上述方法还通过使用基于分布式哈希表(DHT)的无状态路由方法来支持集群重复数据消除；这类似于HYDRAsTOR。

董等[111]提出了两种基于超级块的数据路由算法，其中超级块是一组连续的块。一种是“无状态”路由算法，它使用超级块中的内容(例如超级块中第一个块的前64个字节)来选择一个“bin”，然后将其映射到一个节点。对于负载均衡，箱到节点的映射会随时间而变化。另一种算法是“有状态”路由方法，它利用数据相似性将超级块定向到重复数据消除效果最好的地方。任何负载不平衡都会抵消重复数据消除的好处，因此当前存储的内容超过平均总内容的节点需要相应的重复数据消除改进，才能被选为超级区块的目标。

傅等[112]提出了一种基于相似性和局部性的路由算法，称为Dedupe，该算法在基于相似性分发超级块后，进一步挖掘局部性，以缓解每个节点中的块索引查找瓶颈。Kaiser等人[113]为精确的群集重复数据消除设计一个联合分布式区块索引，但它部分受到节点间通信的限制，节点间通信有严格的带宽要求。生产[114]是一种类似于超块方法的有状态路由方法[111]，它通过数据路由的概率相似性度量，以更少的开销进一步平衡集群节点之间的负载。

总结。通常，基于位置的方法被广泛用于提高重复数据删除索引性能，无论指纹是否存储在硬盘驱动器上[13]或闪存驱动器[61]。基于相似性的方法在减少重复数据删除索引的内存开销方面是有效的[60, 102,

103]。闪存辅助方法会给重复数据消除系统带来额外的硬件成本，但其键值存储方案也可用于索引磁盘指纹[108, 109]。群集重复数据消除方法可针对大规模存储系统进行扩展，但可能会降低重复数据消除率[111, 112, 114]或者需要更多系统资源来保持高重复数据消除率[59, 113]。

### 3.4 重复数据消除后压缩

为了节省空间，重复数据消除已广泛部署在存储系统中。在实践中，基于指纹的重复数据消除方法无法识别显著的冗余部分。每个块通常都有内部冗余，可以用传统的压缩器(如LZ)删除。如果将一起添加到系统中的唯一块一起压缩到一个更大的“压缩区域”，则整体压缩率将高于单独压缩每个块。例如，DDFS报道了一个典型的2压缩[7]。

除了重复数据消除后区块的简单可压缩性之外，仅包含少量不同字节的类似区块之间可能存在高度重叠，例如图中的区块C2和C55。即使只存在几个不同的字节，这些块的基于安全散列的指纹也将完全不同[8, 11, 68]。如第节所述3.1，提出了重新分块方法，通过将非重复数据块进一步分成更小的数据块来提高重复数据消除率[75, 84 - 86]，这有助于识别更多冗余。相比之下，去重复后增量压缩消除了非重复但相似的区块之间的冗余，而不需要重新分块操作来实现更高的冗余消除率[8, 54, 55, 68]。因此，它被认为是一个有效的重复数据消除后流程，可以进一步消除数据冗余，但会增加额外的计算、索引和I/O开销，因此是一个可选的重复数据消除后存储管理阶段(见图4)。

重复数据消除后增量压缩面临的主要挑战来自相似性检测、读取基本区块和增量编码这三个耗时的阶段[54]。桌子6总结了增量压缩中相似性检测和增量编码的最新方法；稍后将讨论回读问题。

相似性检测。在增量压缩中，关键的研究问题是如何以低开销准确地检测出一个相当相似的增量压缩候选。曼伯[123]提出了一种通过计算一组基于多项式的指纹(即Rabin [77])；两个文件之间的相似性与它们之间共有指纹的比例成正比。这种方法已在DERD使用[124]来检测类似的文件，然后

表6  
重复数据消除后增量压缩的最先进的相似性检测和增量编码方法。

| 名字     | 来源                            | 独特的功能                  | 分类学   |
|--------|-------------------------------|------------------------|-------|
| 超级功能   | 《议定书》/《公约》<br>缔约方会议' 00 [119] | 将多个确定性采样散列合并成超特征       | 相似性检测 |
| REBL   | USENIX 04 [120]               | 使用超级特征和增量编码来消除块级冗余     |       |
| 锥形物    | FAST' 05 [51]                 | 使用存储块指纹的布隆过滤器来测量文件相似性  |       |
| 特拉华州   | OSDI 08 [121]                 | 基于子页面的散列来识别相似的页面       |       |
| SIDC   | FAST' 12 [8]                  | 使用内存高效且基于位置的超级功能缓存     |       |
| 敢      | ' DCC' 14 [55]                | 检测唯一但重复的相邻块作为潜在候选块     | 增量编码  |
| xdar   | UCB 理工大学。代表 [49]              | 基于复制/插入指令和 KMP 优化的增量编码 |       |
| Zdelta | NYU 理工大学。代表 [122]             | 将 Zlib 中的霍夫曼编码合并到增量编码中 |       |
| Ddelta | 业绩' 14 [68]                   | 基于基于齿轮的内容定义分块的增量编码     |       |

增量编码。超级功能方法 [119] 是基于布罗德定理 [115] (见等式 7 在部分 3.3): 确定性地采样几个拉宾指纹 (例如, 使用极端宁滨中的最小散列法 [60]) 的文件或块作为特征, 并将它们合并成超级特征, 也称为超级指纹, 然后对超级特征进行索引以检测相似的文件或块。超级特征方法被广泛用于基于增量压缩的冗余消除 [8, 54, 120, 125]。

锥形 [51] 提出了一种超特征方法的替代方案, 即使用记录块指纹的 Bloom 过滤器来表示每个文件, 根据 Bloom 过滤器之间的匹配位数来测量文件相似性, 然后对检测到的相似文件进行增量压缩。差异引擎 [121] 和 I-CASH [126] 充分利用增量压缩技术来分别消除内存页面和固态硬盘缓存中的冗余, 其中它们使用基于计算几个 64B 子页面的哈希的参数化方案来检测相似的 4KB 页面, 这类似于超级功能方法。

基于流的增量压缩 (SIDC) [8] 显示, 在 EMC 经过重复数据消除的备份存储系统之间进行复制时, 重复数据消除后增量压缩可以将数据减少率进一步提高 3/5 (请参见表 2 在部分 2.2)。此外, SIDC 表明, 增量压缩的超级特征索引也会溢出内存容量 (类似于第节中的指纹索引问题 3.3)。SIDC 利用了相似区块的局部性, 这些区块具有针对重复区块所讨论的重复模式, 因此以流通知的方式存储超级功能并将预取到缓存中可以捕获大多数潜在的相似性检测, 同时降低内存需求。SIDC 也适用于存储系统 [54], 它展示了管理增量编码存储的新复杂性。重复数据消除感知相似性检测 (DARE) [55, 127] 通过在重复数据消除之后基于现有的重复邻近信息检测潜在的相似组块以进行增量压缩来扩展这项工作, 即, 如果在重复数据消除系统中两个组块各自的相邻组块被确定为重复, 则认为它们相似。

相似块的增量编码。三角洲压缩面临的另一个挑战是时间-

计算相似数据块之间差异的消耗过程。随着存储容量和网络带宽的快速增长, 增量编码的效率变得越来越重要 [54, 126]。像传统的无损压缩方法一样, Xdelta [122] 使用字节滑动窗口来标识增量计算的基本块和输入块之间的匹配 (即重复) 字符串。这是一个在许多冗余消除系统中广泛使用的开源项目 [8, 55, 121]。Zdelta [68] 结合了霍夫曼编码 [40] 进行增量压缩以进一步消除冗余。最近, Ddelta [68] pro 提出了一种受重复数据消除启发的增量编码方法, 该方法通过快速的基于 Gear 的内容定义分块算法将相似的块分成几个独立且不重叠的字符串。该算法简化并因此加速了增量编码过程, 同时实现了与 Xdelta 和 Zdelta 相当的数据缩减率。

额外的增量压缩挑战。除了上述两个挑战之外, 其他重复数据消除后增量压缩和存储管理阶段, 例如读取基本区块、数据恢复和垃圾收集, 也是重复数据消除后增量压缩的重要方面。具体而言, 压缩相对相似的组块需要读取已经存储的非重复但相似性匹配的组块, 以便与输入组块进行增量编码。Shilane 等人 [54] 建议将唯一区块存储在固态硬盘上, 以解决随机访问磁盘上基本区块导致的增量压缩吞吐量瓶颈。另外, 增量压缩后的数据恢复和垃圾收集仍然是未解决的问题, 将在第节中进一步讨论 4.4。

### 3.5 数据恢复

在识别重复数据并存储非重复数据之后, 希望高效地恢复数据并有效地管理碎片化的存储空间 (在用户的删除操作之后)。后一个过程被称为垃圾收集。因此, 数据恢复和垃圾收集已成为重复数据消除系统存储管理阶段的两个重要问题。本小节主要回顾数据恢复的最新方案, 垃圾收集问题将在下一小节中详细介绍。



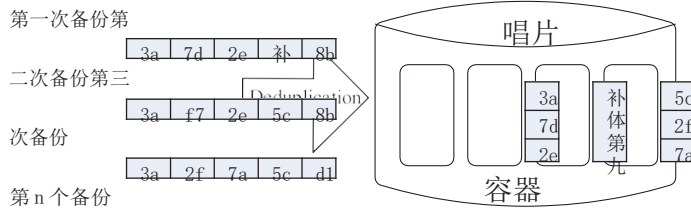


图8. 基于重复数据消除的备份存储系统中的数据碎片示例。请注意，容器是一个固定大小的存储单元，通过使用大型 I/O 来存储顺序的和非重复的区块，以获得更好的备份和恢复性能 [13, 23, 103, 129].

数字8显示了基于重复数据消除的备份存储系统中的数据碎片示例。每个备份（例如，第三次备份）中逻辑上连续的区块在重复数据消除后物理上分散在几个数据容器（固定大小的存储单元）中，也称为区块碎片 [23, 128, 129]，而不是以传统方式紧凑连续地排列。由于硬盘的随机输入/输出性能较差，区块碎片（如磁盘碎片）会显著降低恢复性能。此外，区块碎片还会损害垃圾收集的性能 [62, 129]。例如，如果用户删除了图中的第一个备份8，将很难回收第一次备份中数据块的存储空间，因为其中一些数据块被第二次和第三次备份引用。

一些重复数据消除后的数据恢复方案会重写重复但分散的数据块，以缓解读取（恢复）性能的下降，从而在重复数据消除率（容量节省）和读取（恢复）性能之间进行权衡。桌子7全面研究了在重复数据消除系统中提高恢复性能的最新技术，并根据部署重复数据消除的存储环境将其分为三类，即主存储、备份存储和云存储。

主存储系统对输入/输出延迟敏感 [6, 64] 这使得重复数据消除导致的读取延迟延长碎片问题变得极其重要。iDedup [64] 利用主存储工作负载的空间位置选择性地对连续的重复磁盘块进行重复数据消除，以减少碎片并缓冲随机 I/O 造成的读取延迟。面向性能的 I/O 重复数据消除 (POD) [130] 通过识别关键 I/O 路径中对容量不敏感但对性能敏感的小型重复写入和文件，进一步提高了基于重复数据消除的主存储系统的读取性能。

对于基于重复数据消除的备份存储，在备份系统的整个生命周期中，最近备份的恢复速度可能会下降几个数量级 [23, 129] 由于区块碎片问题。Nam 等人 [128, 136] 建议使用一种称为区块碎片级别 (CFL) 的量化指标来选择性地消除顺序区块和重复区块，该指标类似

至 iDedup [64]。基于上下文的重写 (CBR) [131] 和封顶 [23] 算法使用其特定的碎片度量来确定写入缓冲区（例如 10-20MB）中的碎片块，然后选择性地写入碎片块以提高恢复速度。此外，cap 使用正向组装技术，通过利用恢复和备份时可用的未来块访问的完美已知边缘，来高效地缓存块。

反向重复数据消除（反向重复数据消除）[132, 137] 消除以前备份中的重复数据，而传统的重复数据消除消除新备份中的重复数据。因此，RevDedup 会将碎片转移到旧备份，从而保持最新备份的布局尽可能有序，但代价是后处理重复数据消除。历史意识重写 (HAR) [129, 138] 尝试重写更少的数据，以实现更好的恢复性能，方法是将碎片准确地分为两类：稀疏容器和无序容器，其中容器是 4MB 大小的存储单元，用于存储重复数据消除后的区块。通过利用备份版本的历史信息，从上次备份中识别稀疏容器。作为稀疏容器中块的副本的新块可以被重写以提高数据局部性。因为包含整个文件的区块列表的配方是预先已知的，所以在恢复期间，可以根据 Belady 的最佳缓存方案高效地缓存无序容器 [139]。

对于基于重复数据消除的云存储，恢复速度可能会受到广域网带宽相对较低或频繁访问云服务器中碎片区块的严重限制。基于因果关系的重复数据消除 (CABdedupe) [133] 在客户机和云之间的备份数据集的时间顺序版本中识别未修改的数据，从而提高再存储性能。固态硬盘辅助恢复 (SAR) [134, 140] 在云服务器中的固态硬盘上存储具有高引用计数的唯一区块，以提高恢复性能，这利用了固态硬盘良好的随机读取性能特性。近似精确碎片整理 (NED) [135] 将区块分组为用于云备份的数据段，并根据称为数据段引用率的指标在将数据段上传到云之前识别这些碎片。该方案实现了与没有重复数据消除的云备份系统相当的恢复性能。

总之，重复数据消除系统中的区块碎片问题会导致数据恢复性能下降。最先进的方案在重复数据消除的容量节省和性能损失之间找到了恰当的平衡点。我们认为，在基于重复数据消除的存储系统中，高效、准确的碎片整理可以显著缓解恢复性能的数量级下降 [129]。

表7  
重复数据消除后数据恢复的最新方法。

| 名字                 | 来源               | 独特的功能                     | 分类学         |
|--------------------|------------------|---------------------------|-------------|
| iDedup             | FAST' 12 [64]    | 仅选择性地对磁盘块序列进行重复数据消除(大型写入) | 初级的<br>存储系统 |
| 豆荚                 | IPDPS' 14 [130]  | 专注于对性能敏感但对容量不敏感的小型写入和文件   |             |
| CFL-SD             | 吉祥物 11 [128]     | 基于块碎片级别的度量进行碎片整理(CFL)     | 支持<br>存储系统  |
| CBR                | SYSTOR' 12 [131] | 基于流上下文和磁盘上下文选择性地重复数据消除    |             |
| 盖                  | FAST' 13 [23]    | 基于写缓冲区(约 20MB)分析进行碎片整理    |             |
| 加速                 | APSYS' 13 [132]  | 使用反向重复数据消除优化对最新备份的读取      |             |
| 海港咨询雷达             | USENIX' 14 [129] | 通过利用备份系统的历史信息进行碎片整理       | 云<br>存储系统   |
| CABdedupe          | IPDPS' 11 [133]  | 在按时间顺序排列的数据集版本中识别未修改的数据   |             |
| 特别行政区              | NAS' 12 [134]    | 在固态硬盘上存储具有高引用计数的唯一区块      |             |
| 内德(男子名, 等于 Edward) | ICA3PP' 14 [135] | 备份前基于精确段参考分析碎片整理测试        |             |

### 3.6 碎片帐集

本小节讨论基于重复数据消除的存储系统中最先进的垃圾收集方法。由于唯一的区块可以由多个文件共享, 因此引用管理对于跟踪区块使用情况和回收释放的空间(即垃圾收集)至关重要。一般来说, 垃圾收集包括两个关键步骤, 找到无效的块, 然后回收它们的存储空间。根据第一步, GC 方法通常可以分为两类, 即引用计数和标记-清除。

- 引用计数法及其变体。重复数据消除系统中特定区块的区块引用计数是指该区块被使用/引用的次数。例如, 虽然引用计数为  $N$  意味着该块被引用  $N$  次(在  $1-N$  个文件中), 但是 0 引用计数值意味着该块由于用户的删除操作而不再被共享, 并且可以被回收用于垃圾收集。魏等[104]将这种天真的引用计数方法用于 GC。然而, 为每个块精确地构建一个引用计数器, 这是空间低效的[129]。此外, 郭等[62]指出引用计数可能存在可靠性低的问题: 当出现错误时, 一些块可能会被更新, 而另一些块可能不会。Simha 等人[141]提出一种混合方法, 为每个物理块维护一个参考计数和一个到期时间, 用于增量备份。Strzelczak 等人[142]还维护了一个引用计数器, 并提出了一种基于纪元机制和取消删除标记的并发删除算法, 以分离旧的和新生成的数据, 这允许删除与用户在其商用 HYDRAstor 系统中正在进行的读写同时进行[59]。Dmdedup [143]还将参考计数解决方案用于数据块设备重复数据消除, 但它不会立即回收数据块, 也不会从索引中删除相应的哈希, 从而减少关键写入路径的延迟。傅等[129]提出一种容器标记算法(CMA), 通过仅记录被引用的容器来简化块引用管理, 同时减少每个容器中的碎片块

通过他们的历史感知重写(HAR)算法。

- 标记-清除法及其变体。标记和清除是另一种气相色谱解决方案, 由两个阶段组成。在标记阶段, 所有文件都被转换, 以便标记使用的块。在清除阶段, 清除所有块, 回收未标记的块。分组标记和扫描(GMS)[62]生成一个位图来标记每个容器中的有效块(即图中所示的一组块 8)被每个备份引用, 并通过合并所有容器的位图来定位和回收所有无效区块占用的存储空间, 从而回收存储空间。Botelho 等人[144]构建一个完美的散列向量[145]作为所有区块的紧凑表示, 然后遍历所有文件配方以回收无效区块的存储空间。

总之, 重复数据消除系统中的垃圾收集问题是由重复数据消除后在文件/备份之间共享区块引起的。在重复数据消除系统中, 高效的区块引用管理对于垃圾收集器识别无效区块并回收其存储空间至关重要。参考计数解决方案[104]及其一些变体[129]支持线内 GC 同时标记和清除及其变体[62, 144]本质上是离线的。请注意, 在基于重复数据消除的备份系统中, 垃圾收集通常是在删除一个或多个完整备份后作为后台进程执行的, 但在主存储系统中, 垃圾收集通常是内联或部分内联执行的。一般来说, 在现有的气相色谱方法之间有一个折衷。具体而言, 在基于重复数据消除的存储系统中, 即时垃圾收集会增加关键输入/输出路径的延迟, 而延迟垃圾收集会降低空间利用率。

### 3.7 安全

基于重复数据消除的存储系统面临着固有的严重安全和可靠性问题, 尤其是在云存储系统中, 如 Dropbox、Wuala 和 Mozy [146]。安全问题源于重复数据消除后共享数据块或文件的用户, 这可能会暴露云存储系统中的安全漏洞和隐私问题。

目前, 跨用户重复数据消除面临以下三大安全挑战。

- (1) 与 en-的矛盾

加密:不同的用户可以用他们自己的密钥加密他们的数据。在这些情况下,不同用户的相同数据将导致不同的密文,从而无法在不同用户之间进行重复数据消除。(2)侧信道攻击:跨用户去重发生可以通过识别文件、学习文件内容、建立隐蔽信道[147].

(3)所有权证明:存在一种攻击,使用一个小的哈希值(即指纹)作为整个文件的代理,客户端可以向服务器证明它确实上传了文件。具体来说,知道文件的代理指纹的攻击者可以说服存储服务器它拥有该文件,这样服务器将允许攻击者下载整个文件[148].

桌子8根据上述三个挑战,检查并分类保护重复数据消除的最先进方法,下面将详细介绍。

- 收敛加密(CE)使用从数据内容获得的散列作为密钥来加密数据。因此,来自不同用户的相同数据将生成相同的密文,这使得加密后的重复数据消除可以跨不同的用户工作。1995年,备份系统专利首次描述了将基于内容的键控和重复数据删除相结合的方法[149],然后在微软的Farsite分布式文件系统中被开发并称为聚合加密[56, 150]. CE广泛用于许多重复数据消除系统[151 - 154]. 消息锁定加密[155]研究和比较CE及其变体,基于哈希的CE[151],在实践和理论上随机化CE。为了进一步确保消费电子产品的安全性,双重[156]通过一个不经意的RSA-OPRF协议(基于RSA的阻塞伪随机函数)从密钥服务器添加了一个ce密钥实例,该协议可以很好地抵抗来自传统终端CE的暴力攻击。为了确保CE密钥的安全性和可靠性,Dekey通过Ramp秘密共享方案(RSSS)将要共享的CE密钥分发到多台服务器,以实现高效的CE密钥管理[154]. SecDep结合了跨用户文件级和用户内部块级重复数据消除,并在用户之间和内部使用不同的ce变体来最小化计算开销。此外,SecDep使用文件级密钥来加密块级密钥,因此密钥空间不会随着共享用户的数量而增加。
- 侧信道攻击是指由云存储服务器执行的多客户端重复数据消除,其中一个客户端站点对手使用重复数据消除作为侧信道攻击,以揭示关于其他客户端的文件/块的内容的私有信息(即,具有相同和敏感的内容)。Harnik等人[147]提出一种混合方法,有时会人为关闭跨用户重复数据消除,以降低数据泄露的风险。海伦

等人[157]提出一种基于网关的重复数据消除方法,该方法允许从网关到服务器而不是从客户端进行重复数据消除,从而抵御来自客户端对手的侧信道攻击。

- 所有权证明(POW)允许客户端有效地向服务器证明他/她持有文件,这用于防止攻击者根据文件的指纹访问其他用户的文件内容。Halevi等人[148]首先引入POW的概念,允许客户端通过构造纠错码和Merkle树(用于数据认证)来有效地向服务器证明客户端拥有文件[158]的文件。随后提出了功率的变体,以减少产生功率的开销[159]或进一步确保POW上的安全性[160, 161].

此外,李等人还构建了一个名为CDStore[163]最近,为了确保重复数据消除的数据安全性和可靠性。CDStore不同于传统的CE方法,它提出了一种称为CAONT-RS的增强秘密共享方案[164],它用确定性加密哈希信息替换原始随机信息,并将数据拆分为 $n$ 个秘密份额。因此,相同的数据将生成相同的秘密共享,从而在秘密共享上实现重复数据消除。CDStore还通过将秘密份额分发到多云中并通过获得 $k$ 个秘密份额(在CAONT-RS中为 $k < n$ )来恢复数据来确保可靠性,但在重复数据删除节省的存储空间和使用CAONT-RS增加的存储成本之间存在权衡。

现有的安全重复数据消除方法侧重于解决跨用户重复数据消除中的三个已知安全挑战,尤其是在云存储中。关闭跨用户重复数据消除将提供最高的安全级别,以避免用户之间的隐私泄露[147, 161]. 但是,用跨用户重复数据消除来换取数据安全仍然是云存储面临的一个挑战,这将在本节中作为未来方向之一进行讨论4.4。

### 3.8 可靠性

重复数据消除降低了存储系统的可靠性,因为丢失几个关键数据块会导致许多被引用的文件丢失[165, 166]. 此外,由于重复数据消除减少了所有现有可靠性和容错机制所依赖的冗余,可靠性问题可能会进一步出现。例如,故障存储设备中的数据恢复是通过利用通过添加副本(例如,RAID中的复制或奇偶校验)引入的数据冗余来实现的,如果重复数据消除完全消除了冗余数据,这些副本将无法工作。

一般来说,最先进的方法可以大致分为两类,重复数据消除,然后是“RAID”和基于引用计数的复制,如下所述。

表 8  
保护重复数据消除的最先进方案。

| 名字        | 来源                   | 独特的功能                        | 分类学      |
|-----------|----------------------|------------------------------|----------|
| 道格拉斯等人。   | 专利(1995 年) [149]     | 在备份系统专利中首次提出了融合加密            | 收敛加密及其变体 |
| Farsite   | ICDC ' 02 [150]      | 为文件级重复数据消除开发聚合加密 (CE)        |          |
| 模仿作品      | OSDI ' 02 [151]      | 为块级重复数据消除提出基于哈希的 CE (HCE)    |          |
| 最大似然估计    | 欧洲墓穴' 13 [155]       | 提出随机收敛加密 (RCE)               |          |
| 双重        | USENIX 党卫军' 13 [156] | 引入服务器辅助的抵抗暴力攻击的 ce           |          |
| Dekey     | IEEE TPDS' 14 [154]  | 使用斜坡秘密共享方案进行消费电子密钥管理         |          |
| SecDep    | MSST' 15 [162]       | 提出用户感知消费电子和多级密钥管理            | 侧通道攻击    |
| Harnik 等人 | IEEE S&P' 10 [147]   | 有时会人为关闭跨用户重复数据消除             |          |
| 基于网关      | IEEE NTMS' 12 [157]  | 重复数据消除发生在服务器和网关之间，而不是客户端     |          |
| 战俘        | CCS' 11 [148]        | 引入 POW，并通过 Merkle 树和特定编码进行解析 | 所有权证明    |
| s-POW     | 亚洲 CCS' 12 [159]     | 使用随机选择的位串的知识作为文件的功率          |          |
| 徐等。       | 亚洲 CCS' 13 [161]     | 在弱泄漏弹性设置中实现功率                |          |

- 重复数据消除“RAID”是指在重复数据消除之后，使用特定的擦除代码直接对唯一的区块进行编码。DDFS [13]使用软件 RAID—

6 存储独特的区块，以确保高水平的数据完整性和可靠性。Hydrastor [59]将独特的数据块放入不同的弹性类中，每个类都有用户提供的不同级别的可靠性。在 R-ADMAD [167]，可变大小的块首先被打包成更大的固定大小的对象 (例如，8MB)，然后这些对象被擦除编码并分布在冗余组中的多个存储节点上。此外，将唯一区块复制到另一台服务器通常也是确保存储可靠性的一种简单而有效的方法，因为数据已首先进行了重复数据消除 [168]。

- 基于引用计数的复制根据唯一块的引用计数提供不同级别的可靠性。Bhagwat 等人 [165] 请注意，由于共享公共区块，重复数据消除会改变存储数据的可靠性。他们通过从复制级别选择每个块来解决这个问题，复制级别应该与该块的受欢迎程度成对数关系。Rozier 等人 [169] 设计并实施建模框架，以评估具有不同硬件和软件配置的重复数据消除系统的可靠性。基于引用计数的复制方法是确保重复数据消除系统可靠性的首选方法。

在设计可靠的重复数据消除时，根据唯一区块的参考计数利用不同级别的可靠性有助于提高基于重复数据消除的存储系统的整个系统可靠性。但是，通过重复数据消除在确保可靠性和节省空间之间始终存在权衡。此外，李等 [166] 还建议，如果使用传统的 MTDL 指标来评估可靠性，则重复数据消除可能会提高存储系统的可靠性。这是因为重复数据消除减少了所需磁盘驱动器的数量，从而降低了发现磁盘错误的可能性。

### 3.9 其他有趣的问题

重复数据消除中还有其他有趣的问题，例如重复数据消除率估计、文件配方压缩以及视频/图像重复数据消除等。，这是未来研发关注的问题。

- 重复数据消除率估计的研究受到以下观察的启发，即重复数据消除率可能根据工作负载和底层重复数据消除技术而显著变化 [170–173]。具体数据集中重复率的准确估计将允许用户决定购买多少磁盘、使用什么技术等。Harnik 等人 [173] 使用采样和扫描的通用框架来估计重复数据消除率，以减少时间和内存开销。谢等 [172] 提出一种自适应技术，以增量方式维护重复数据消除率的最新估计值，该估计值考虑到文件系统的任何更改。
- 文件配方压缩是由 Meister 等人提出的 [174] 通过记录区块指纹序列来进一步减少文件配方的空间开销。他们报告称，对平均数据块大小为 8KB、SHA-1 指纹和重复数据消除系数为 30 的 1 PB 备份数据集进行重复数据消除，将生成约 2.4 TB 的文件配方，使用零数据块抑制和熵编码等技术，可以有效减少高达 90% 的文件配方。
- 视频和图像重复数据消除根据视频/图像的视觉内容检测并消除非常相似的视频/图像之间的冗余，这与传统的重复数据消除方法有很大不同，但最近在学术界和工业界受到越来越多的关注。参见 [175] 提出了一个通用框架，用于从应用程序级别的角度检测非常相似的视频，以消除冗余。Perra 等人 [176] 利用最先进的视频压缩技术，在几个非常相似的图像之间高效地重用图像内容，这种技术被称为图像内容重复数据消除，以支持相似图像的大规模压缩。通过使用语义哈希和平面寻址，FAST [177] 探索和开发

图像的语义相关性，快速识别相似内容，支持近实时图像检索。Dewakar 等人 [178] 指出类似的视频文件可以从他们提出的内容感知重复数据消除技术中显著受益，在某些使用情况下，该技术可将存储效率提高高达 45%。

## 4 资源和前景

在本节中，我们将首先研究重复数据消除的典型应用场景，以及它们如何融入重复数据消除并从中受益。然后，我们提供了重复数据消除在商用存储系统中如何工作的行业见解。我们还介绍了重复数据消除研究的公开可用资源，包括开源项目、数据集、跟踪等。最后，我们通过确定研究社区面临的开放问题和研究挑战作为重复数据消除的未来可能方向来结束本节。

### 4.1 重复数据删除的应用场景

重复数据消除方案已经广泛应用于计算机系统的许多场景中。桌子 9 介绍了重复数据消除的一些典型应用场景及其有益功能。

- 二级存储。二级存储系统中有大量的副本，如备份和归档存储 [7, 11, 13, 58, 59, 187]。基于这一观察，数据域 [13]，一家存储公司，现在是 EMC 的一部分，认为“基于磁盘的重复数据消除存储已经成为新一代企业数据保护存储系统，取代了磁带库。”事实上，在此类系统中使用重复数据消除已被证明可实现约为的数据缩减系数 5 40 [7, 8]，从而大幅节省存储空间和相应的硬件成本。最近，重复数据消除后增量压缩被用于压缩不重复但相似的数据块，作为重复数据消除的补充方法。这种后重复数据消除方案可实现额外的数据缩减系数 2 5 在重复数据消除之上，但增加了额外的计算和输入/输出开销 [54, 55, 68]。
- 主存储器。最近的研究表明，重复数据消除可以在主存储器中实现高达 40–60% 的数据缩减率 [5]，尤其是对于服务器文件系统 [5, 6, 179] (参考表 2)。主存储重复数据消除不仅降低了存储空间需求，还消除了关键输入/输出路径上的重复输入/输出 [64, 130, 179]，这有助于提高主存储的磁盘 I/O 性能。最近，一些用于主存储的开源文件系统，如 ZFS [73]，开放式重复数据删除 [188] 和 Lessfs [189]，具有企业级重复数据消除功能，以获得更好的存储性能。

由于重复数据消除会导致写入路径中的计算和索引延迟以及读取路径中的磁盘碎片，因此 ZFS 和 Lessfs 将重复数据消除作为可选功能提供给用户。请注意，与辅助存储重复数据消除相比，主存储重复数据消除有三个主要原因：输入/输出延迟对重复数据消除操作更敏感，读取和删除操作发生得更频繁，主存储系统中的重复数据减少。

- 云存储。近年来，云存储已成为计算机系统的重要存储平台 [74, 180]。由于底层广域网中有限的网络带宽是云存储的主要性能瓶颈，因此重复数据消除可以通过识别未修改的数据来帮助加速客户端和云之间的数据同步。同时，重复数据消除还有助于降低云端的存储开销。目前有 DropBox、SkyDrive (现在叫 OneDrive)、Google Drive 等。整合重复数据消除以提供更好的云存储服务 [74, 190]。正如微软的研究中所讨论的 [5]，虽然跨用户存在大量重复数据，但跨用户重复数据消除会导致安全问题 [161] (见第 3.7)。这仍然是一个悬而未决的问题，将在第 4 节中进一步讨论 4.4。
- 虚拟机。虚拟机中存在大量冗余数据，无论是在主存储器中 [121, 191] 或外部存储器 [192]。这是因为同构或异构虚拟机上的操作系统和应用程序往往会生成重复数据。此外，通过节省两个存储器的存储空间，重复数据消除满足了计算机系统中虚拟化的设计目标 [121, 181, 191, 193] 和磁盘 [182, 192, 194]，从而减轻存储设备的负担。此外，在许多最先进的方法中，重复数据消除通过显著减少迁移的数据量来加速虚拟机的实时迁移 [15, 195, 196]。
- 网络环境。使用重复数据消除的最初目的之一是通过避免传输冗余数据来节省网络带宽，尤其是在广域网 (WAN) 环境中 [89, 197–199] 和航空电子网络环境 [200]。网络重复数据消除，也称为冗余消除，与存储系统中的重复数据消除略有不同。具体来说，网络重复数据消除的粒度 (即数据块的大小) 通常为几十或几百字节，比备份存储重复数据消除中的 KB 级 (甚至 MB 级) 粒度要精细得多。此外，网络环境中数据重复的对象是数据流或数据包，网络重复数据消除通常对手指使用较弱但较快的散列



表9  
重复数据消除的典型应用场景及其最著名的示例。

| 情节        | 重复数据消除的优势            | 例子                    |
|-----------|----------------------|-----------------------|
| 辅助存储器     | 节省存储空间、设备成本、备份/归档时间  | Venti [111359]        |
| 主存储器      | 节省存储空间，消除重复的磁盘输入/输出  | 输入/输出重复数据消除[1796473]  |
| 云存储       | 节省存储空间，减少广域网上传数据的时间  | 积云 1807474            |
| 虚拟计算机     | 节省主/主存储空间，节省虚拟机迁移时间  | VMware ESX [18115182] |
| 网络        | 为广域网优化减少传输冗余的时间      | Spring 等人[18318481]   |
| SSD、多媒体等。 | 延长固态硬盘的使用寿命，消除视觉冗余等。 | CAFTL [185186175]     |

以逐字节方式识别数据冗余的打印算法。微软研究院的一项研究[89]显示包级冗余消除技术可以在12个大型网络节点上实现约15-60%的带宽节省。与存储系统不同，用于网络传输的重复数据消除没有碎片化的局部性和垃圾收集的开销，因为传输的数据通常在目标位置重建。

- 耐久性受限的非易失性存储。最近，重复数据消除已被用于新兴的非易失性存储设备，这些设备具有耐久性限制(即写入限制)，例如固态硬盘设备，以减少设备的写入流量并增加其有效逻辑容量。CAFTL [185]和CA-SSD [201]采用重复数据消除来消除重复的输入/输出，以延长固态硬盘的使用寿命。I-CASH [126]使用增量压缩来消除重复和相似的数据，以扩大固态硬盘缓存的逻辑空间。硝基[186]为基于固态硬盘缓存的主存储实施重复数据消除和LZ压缩，以降低固态硬盘缓存的成本并增加逻辑容量。
- 其他应用。最近，非常相似的多媒体文件中的重复数据消除越来越受到关注，如第节所述3.9。Katiyar等人[175]和Perra等人[176]基于非常相似的视频/图像的视觉内容而不是二进制内容来识别和减少它们之间的冗余。Dewakar等人[178]建议在某些使用情况下，重复使用类似的视频可以将存储效率提高45%。唐等[202]介绍UNIC，这是一个安全地重复一般计算的系统，它通过记忆和重用计算结果显著地加速了一些应用程序。

总的来说，重复数据消除在备份/归档存储系统中的应用最为广泛，并开始在主存储中采用。通过消除重复数据，重复数据消除不仅为备份/归档存储系统提供了许多好处，还为其他应用场景提供了好处，例如减少主存储的重复输入/输出、避免为网络环境传输重复数据、延长新兴非易失性存储设备的寿命等。随着大数据时代数字数据量的爆炸式增长，我们相信会有更多的ap-

受益于重复数据消除的应用程序，可有效识别和消除冗余数据的重复拷贝，从而提高系统性能。

## 4.2 重复数据消除的行业前景

自2004年以来，重复数据消除已在专用备份设备中投入商用[168]，多家供应商已开发出竞争产品[59, 203-205]。从那时起，重复数据消除已成为从静态归档到高性能主存储的众多存储系统的标准功能。虽然重复数据消除已得到广泛实施，但我们将讨论的重点放在备份存储上，因为它的历史最长，而且它自2004年以来的发展让我们深入了解了该行业如何适应技术变化。自2004年引入重复数据消除备份存储以来，该市场在2013年增长至31亿美元[206]。

### 4.2.1 从磁带转移到磁盘

早期的备份被写入具有高顺序读/写性能的磁带系统，但它们也有其他使恢复变得复杂的属性。首先，磁带必须在使用前明确安装；即使这些挂载从人工操作发展到自动“点唱机”操作，每个挂载仍然遭受显著的延迟。第二，由于磁带的硬件特性，它的随机IOPS非常低。出于这些原因，磁带备份传统上遵循一种模式，即系统的全部内容将被一次备份(“完整”备份)，然后自上次完整备份以来对系统的更改将被定期保存。后者称为“增量”备份，包括文件中任何地方修改过的每个文件。

由于性能差异，从磁带到硬盘的转换为客户提供了许多优势，但是单独的硬盘存储太贵了，无法在逐字节的基础上进行成本竞争。但是，每次将完整备份写入系统时，它都会包含大量未全部或部分修改的文件。重复数据消除与压缩相结合，成为提高有效容量和降低系统成本的一种有前途的方法。此外，重复的数据块往往以连续的顺序写入，从备份到备份基本没有变化。这

利用高重复率内容和持续模式的组合来创建高性能、经过重复数据消除的备份存储。

一旦数据在硬盘上进行了重复数据消除，其处理方式可能与存储在磁带上的方式不同。数据可以在本地或异地快速复制，因为经过重复数据消除的数据只是其原始大小的一小部分。与发送到异地的磁带相比，数据的准确性可以得到快速和定期的验证，并且由于每个新版本相对于其原始大小只需要很少的存储空间，因此可以在现场保存更多的版本。

硬盘的输入/输出特性不同于磁带，因此公司和客户开始以新的方式使用系统。主存储发送已更改数据的效率更高，经过重复数据消除的系统可以重建完整备份，而不是发送主要由重复内容组成的完整备份。此过程将增量备份（可能每天、每小时或更频繁）传输到备份服务器，在那里合成完整备份。只需要传输已修改的文件或已更改的数据块，然后通过创建一个引用新数据和先前写入的重复内容的方法来合成完整备份。经过重复数据消除的系统上的磁盘表示与写入完整备份基本相同，因为两种备份方式都只存储唯一的数据块，以及表示文件的方法。主存储系统上的输入/输出和中央处理器负载减少，因为读取的数据更少，传输的数据更少，网络流量也更少。

从

(a) 写入完整备份，然后对其进行重复数据消除，(b) 永远写入增量备份，只需编写一个引用备份中包含的所有区块的新文件配方，即可合成完整备份。当写入新的完整备份时，重复数据消除系统可以选择将新备份中的重复数据写入磁盘，然后垃圾收集重复数据块的旧副本。通过将新备份的连续数据块一起写入磁盘，备份的后续恢复将经历更少的随机输入/输出，并具有更好的读取性能。如第节所述 3.5，有各种技术来决定何时编写新的副本。但是，如果工作负载仅由增量更新组成，导致恢复期间出现更多随机输入/输出，则系统必须更加主动地重组文件系统，以提供可接受的读取性能。

#### 4.2.2 弥合主存储和备份存储的鸿沟

主存储和备份存储过去是完全不同的系统，运行备份软件的媒体服务器充当中间件来协调备份传输。为了从增量更改的数据中有效地合成完整备份，需要一个支持重复数据消除的新界面。这激发了

备份和主存储系统之间更紧密的集成。备份代理直接在主存储上运行并直接备份而不通过单独的媒体服务器（如 DDBOOST [207] RMAN [208]，网络备份 [209] 和 Avamar [210]）。在某些情况下，这会增加主存储系统上的 CPU 负载，同时允许备份系统处理更多的并发写入。在其他型号中，备份存储可能作为可装载到主存储的设备或文件系统出现。经过重复数据消除的备份存储不再局限于磁带替换，而是成为存储层次结构中的另一层。虽然经过重复数据消除的备份存储仍然比主存储慢，但对更高性能的需求继续增长。

客户希望他们的备份比静态内容更有用，静态内容只在数据丢失的极少数情况下才被读回。虽然备份应该是写保护的，但是在一些情况下，客户可能希望读取数据。例如，在数据丢失的情况下，客户可能希望浏览他们在备份存储上的文件，并查看几个文件来确定哪个版本最有用。作为第二个例子，当对同时服务客户端请求的活动版本执行相同的分析不切实际时，可以对存储在备份服务器上的数据库执行数据分析。在某些情况下，客户的主存储系统可能会长时间无法运行。在调配和配置新的主存储硬件时，客户可能希望从备份存储运行应用程序。客户通常会复制备份，将其标记为可写，然后积极使用该版本。虽然备份系统可能不具备主存储的性能，但它可能足够快，可以为有限的工作负载提供服务。例如，虚拟机可以将备份存储用作虚拟机映像的数据存储。第二个例子是，在升级主版本之前，客户可能希望在从备份存储运行的系统上测试软件升级的不兼容性。

在十多年的时间里，经过重复数据消除的备份存储已经从磁带替代发展为一层存储系统，可以充当有限的主存储。虽然我们已经从用例的角度对此历史进行了简要的总结，但是需要大量的技术改进来实现这些变化，包括本次调查中讨论的性能和管理改进。

### 4.3 开源社区中的重复数据消除

在这一小节中，我们将列出一些公开可用的开源项目、数据集和数据重复的跟踪，它们对于研究社区来说是一种有用且有价值的资源。

- 开源项目。LBFS 是最著名的开源项目之一<sup>1</sup>，它实现了

1. <http://pdos.csail.mit.edu/lbfs/index.html>

网络文件系统中基于 CDC 的重复数据消除 [14]. 打开<sup>2</sup>是一个基于重复数据消除的文件系统, 旨在为数据块设备上的应用程序提供线内重复数据消除和灵活性 [188]. 外勤人员<sup>3</sup>是美因茨大学开发的工具 [57, 70, 174] 这使得研究人员能够分析文件系统目录的内部和时间冗余. Destor<sup>4</sup>是一个指纹索引和恢复的应用评估平台, 已在多篇研究论文中使用 [103, 129, 211]. 此外, Dmddup<sup>5</sup>是一个实用的主存储重复数据消除平台, 在数据块层运行重复数据消除 [143] 和 Lessfs<sup>6</sup>是一个用于 Linux 的内嵌数据重复数据消除文件系统 [189].

#### • 数据集、数据跟踪和基准。这

大型开源项目的档案(即标记的源代码), 如 Linux<sup>7</sup> 海湾合作委员会<sup>8</sup>等等, 已在许多研究论文中针对重复数据消除进行了评估 [55, 60, 75, 129]. 虚拟机映像<sup>9</sup>, 如 Ubuntu、Fedora 等, 也已经过评估 [68, 132, 134, 192]. 金融情报室的教学大纲 [179] 已发布输入/输出跟踪<sup>10</sup>(包括每个输入/输出的内容哈希值)持续三周。它的总大小约为 1TB, 主要用于评估几种主要的重复数据消除方案 [130, 201, 212]. 石溪大学的文件系统和存储实验室 (FSL) 已经公布了他们从几个研究生的主目录和 MacOS 快照中收集的痕迹, 总大小为几兆字节<sup>11</sup>, 并开发了一个工具来根据重复数据消除跟踪重新生成数据集 [213]. 此外, 还有几个基准来生成用户文件的工作量 [190, 214] 和可压缩内容 [215] 可用于重复数据消除分析。

#### 4.4 未解决的问题和未来的方向

尽管重复数据消除已经研究了十多年, 但许多公开的问题和挑战仍有待解决, 尤其是随着数字数据的规模继续呈指数级增长, 对长期存储管理的需求变得越来越迫切。基于第节中讨论的研究 3 和科 4, 在接下来的内容中, 我们将概述重复数据消除的未解决问题和未来可能的发展方向。

- 通过散列比较, 还是不比较? 主流的重复数据消除系统, 如 LBFS [14], Veti [11] 和 DDFS [13], 使用基于 SHA1 的指纹来表示和识别数据块, 这是合理的, 因为任意两个随机生成的块之间的加密散列冲突的可能性比许多硬件错误的可能性小许多数量级 [11]. 汉森 [65] 指出, 基于哈希的比较方法并不是没有风险的, 在长期使用的存储系统中, 很难发现和修复重复数据消除的哈希冲突所导致的错误和 bug。Black 等人 [66] 认为在 2006 年发现一次 SHA1 碰撞将花费大约 8000 万美元和 2 年时间, 因此提出哈希比较是完全合理的。因此, 一些系统通过检查内容来解决这个问题 [60, 187] 经过哈希比较后。其他一些系统选择使用更强的加密散列, 例如 SHA-256 [73, 74], 用于指纹识别。
- 重复数据删除传输标准。随着商用重复数据消除存储系统的激增, 提高存储效率的下一个潜在阶段是创建一个支持重复数据消除的传输协议。我们简要总结了以下几种目前可用的 API。Symantec OpenStorage [216] 大约于 2007 年推出, 为基于磁盘的备份平台与赛门铁克网络备份的集成提供了一个 API [209]. 2010 年推出的 Data Domain 的 DDBOOST 应用编程接口同样将众多备份软件系统与 Data Domain 应用装置集成在一起 [207]. 通过提供一个插件, 备份软件可以在其中执行分块和指纹识别, 并允许备份应用装置确定哪些块是重复的, DDBOOST 分配处理, 并且当重复率很高时, 还减少了网络负载。这种仅传输唯一数据的技术能否获得更广泛的适用性? 我们已经看到基于配方的内容分发系统, 如 Bittorrent [217]. 随着越来越多的基础设施以重复数据消除、内容寻址文件系统的形式可用, 像 Bittorrent 这样的利基应用程序之外的网络传输将适应为数据而不是数据本身提供配方。有了这样的协议, 就有可能将数据从一个经过重复数据消除的产品传输到另一个经过重复数据消除的产品, 而不会在传输过程中使数据膨胀。
- 重复数据消除后的恢复和垃圾收集。重复数据消除会导致顺序或连续数据碎片化, 这是主存储和辅助存储中重复数据消除后恢复和垃圾数据收集过程面临的一个严重问题, 因为数据块是共享的。现有解决方案试图通过不进行重复数据消除来消除碎片(即,

2. <https://code.google.com/p/openedup/>

3. <https://code.google.com/p/fs-c/>

4. <https://github.com/fomy/destor>

5. <http://git.fsl.cs.sunysb.edu/linux-dmddup.git/>

6. <http://www.lessfs.com/wordpress/>

7. <ftp://ftp.kernel.org/>

8. <http://ftp.gnu.org/gnu/>

9. <http://www.thoughtpolice.co.uk/vmware>

10. <http://syllab-srv.cs.fiu.edu/doku.php?>

\$id=projects:iodedup:start

11. <http://tracer.filesystems.org/>



重写)一些重复但片段化的块[23, 129], 这牺牲了重复数据消除效率, 并且不能完全解决长期主存储系统或辅助存储系统的数据恢复和垃圾收集问题。此外, 将重复数据消除与(重复数据消除后)增量压缩和传统压缩技术相结合会增加这些问题的复杂性, 这是因为恢复增量压缩数据块需要读取基本和增量数据块, 这是两个磁盘 I/O, 而不是一个。

- 安全的重复数据消除是用户最关心的问题之一, 尤其是在云环境中。跨用户重复数据消除可能会泄露用户的机密信息[147]或者通过获取代表数据内容的指纹而导致一些可能的新攻击[148]。据我们所知, 虽然大多数现有的研究都集中在重复数据消除技术的安全性上, 但没有已发表的研究讨论数据压缩和重复数据消除联合使用时的安全性问题。事实上, 在当前的备份存储系统中, 数据压缩是在重复数据消除之后实现的[7, 13], 而经典的收敛加密方法[150]首先对数据进行加密, 然后在密文上计算指纹进行重复数据消除。这是矛盾的, 因为加密使数据随机化, 因此不可压缩。此外, 保证删除[144, 218]也是基于重复数据消除的云环境中一个很有前途的安全主题。
- 可靠性是基于长期重复数据消除的存储系统的另一个重要问题。复制具有高引用计数的唯一区块的现有方法是解决重复数据消除可靠性问题的好办法。看看如何将方法与数据恢复和垃圾收集过程相集成, 以便在重复数据消除后获得更好的存储性能, 这将是很有趣的。此外, 对已消除重复数据的长期可靠性分析在长期主存储系统或辅助存储系统中非常重要, 但这种分析仍然缺乏。
- 主存储重复数据消除。随着应用程序数据的快速增长, 重复数据消除在主存储系统中越来越受欢迎。采用重复数据消除来提高输入/输出性能[130]将在高性能存储系统中发挥比节省存储空间更重要的作用。读取和删除操作经常发生在主存储系统中。在具有不同存储设备(如磁盘、动态随机存取存储器、非易失性存储设备)的系统中, 利用可变的主存储工作负载来解决这些问题将是一个有趣的未来研究方向。
- 新兴应用。如第节所述 4, 重复数据消除可以使磁盘存储以外的应用程序受益, 例如将重复数据消除用于

延长固态硬盘和相变存储器的寿命[185, 201]并消除图像和视频的视觉冗余[175, 176]。我们相信重复数据消除将会有更多的应用, 例如磁带存储系统[219]或叠瓦式磁盘, 因为这将有助于减少大规模存储系统中不断增长的冗余数据。

## 5 摘要

重复数据消除是大规模存储系统的一种可扩展且高效的冗余数据减少技术, 可应对数据存储容量需求急剧增长带来的挑战。在这项全面的调查研究中, 我们回顾了重复数据消除的背景以及重复数据消除与传统数据压缩的区别。我们还全面研究了最先进的重复数据消除工作, 根据重复数据消除 workflow 将它们分为六大类, 然后为每一类创建一个分类, 从而深入了解现有解决方案的优缺点。本文还深入研究了使用重复数据消除的应用。此外, 为了便于研究团体进一步研究和开发, 对公开的开源项目、数据集和跟踪进行了总结。最后, 根据我们对现有工作和重复数据删除技术应用的综合研究, 概述了存在的问题和研究挑战。

## 感谢

我们感谢匿名审稿人的宝贵意见和反馈。该工作获得国家自然科学基金委 61502190、61232004 部分资助; 863 项目 2013AA013203 计算机体系结构国家重点实验室, 编号: CARCH201505HUST 中央大学基础研究基金, 批准号: 2015MS073CNS- 111660 和 CNS-1016609 资助下的美国国家科学基金会; 教育部信息存储系统重点实验室; 和 EMC。我们还感谢索娜姆·曼达尔、杰夫·库恩宁、瓦西里·塔拉索夫和埃雷兹·扎多克对重复数据消除存储文献的宝贵讨论。

## 参考

- [1] “数据泛滥,” <http://www.economist.com/node/15579717>.
- [2] “IDC. 2011 年数字宇宙研究。技术报告。” <http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf>, 2010 年 6 月。
- [3] J. Gantz 和 D. Reinsel, “2020 年的数字世界: 大数据、更大的数字阴影和远东最大的增长”, IDC iView: IDC 分析未来, <http://www.emc.com/collateral/analyst-reports/idc-digital-universe-united-states.pdf>, 2012.
- [4] “机遇的数字宇宙: 丰富的数据和物联网日益增长的价值,” <http://www.emc.com/leadership/digital-universe/2014iView/executive-summary.htm>, 2014 年 4 月, EMC 数字世界, 由 IDC 进行研究和分析。

- [5] D. 迈耶和沃·博洛斯基, “实用重复数据删除研究”, 载于《USENIX 文件和存储技术会议论文集》(FAST' 11)。美国加州圣何塞:USENIX 协会, 2011 年 2 月, 第 229-241 页。
- [6] A. 埃尔-希姆, r. 卡拉奇, a. 库马尔等, “主要重复数据删除-大规模研究和系统设计”, 在 2012 年 USENIX 年度技术会议 (USENIX' 12) 的会议记录中。美国马萨诸塞州波士顿:USENIX 协会, 2012 年 6 月, 第 1-12 页。
- [7] G. Wallace, F. Douglass, H. Qian 等, “生产系统中备份工作负载的特征”, 载于第十届 USENIX 文件和存储技术会议记录 (FAST' 12)。加利福尼亚州圣何塞:USENIX 协会, 2012 年 2 月, 第 1-14 页。
- [8] 页 (page 的缩写) Shilane, M. Huang, G. Wallace 等, “使用流信息增量压缩对备份数据集进行广域网优化复制”, 载于第十届 USENIX 文件和存储技术会议记录 (FAST' 12)。美国加州圣何塞:USENIX 协会, 2012 年 2 月, 第 1-14 页。
- [9] 长度 DuBois, M. Amaldas 和 E. Sheppard, “重复数据删除向主存储发展的关键考虑因素”, 第 223310 卷白皮书, <http://www.bredrock-tech.com/wp-content/uploads/2010/05/wp-key-considerations.pdf>, 2011 年 3 月。
- [10] W. 博洛斯基, 科尔宾, 戈贝尔等, “视窗 2000 中的单实例存储”, 载于第四届 USENIX 视窗系统研讨会论文集。美国华盛顿州西雅图市:USENIX 协会, 2000 年 8 月, 第 13-24 页。
- [11] 南昆兰和多沃德, “文蒂:档案存储的新方法”, 载于《USENIX 文件和存储技术会议论文集》(FAST' 02)。美国加州蒙特利:USENIX 协会, 2002 年 1 月, 第 1-13 页。
- [12] C. Policoniades 和 I. Pratt, “检测存储系统数据冗余的替代方案。” USENIX 年度技术会议论文集, 普通轨道。美国马萨诸塞州波士顿:USENIX 协会, 2004 年 6 月, 第 73-86 页。
- [13] B. 朱, 李, 帕特森, “避免数据域重复数据删除文件系统中的磁盘瓶颈”, 在第六届 USENIX 文件和存储技术会议记录 (FAST' 08) 第 8 卷中。美国加州圣何塞:USENIX 协会, 2008 年 2 月, 第 1-14 页。
- [14] A. “一个低带宽的网络文件系统”, 载于 ACM 操作系统原理研讨会论文集(' 01)。加拿大温哥华:美国计算机学会协会, 2001 年 10 月, 第 1-14 页。
- [15] 南阿尔-基斯万尼, D. Subhraveti, P. Sarkar 和 M. Ripeanu, “VM- Flock:虚拟机对云的协同迁移”, 载于《第 20 届高性能分布式计算国际研讨会论文集》(HPDC' 11)。美国加州圣何塞:美国计算机学会协会, 2011 年 6 月, 第 159-170 页。
- [16] J. 齐夫和伦佩尔, “顺序数据压缩的通用算法”, IEEE 信息论学报, 第 23 卷, 第 3 期, 第 337-343 页, 1977 年。
- [17] ——“通过可变速率编码压缩单个序列”, IEEE 信息论交易, 第 24 卷, 第 5 期, 第 530-536 页, 1978 年。
- [18] 米 (meter 的缩写) Oberhumer, “LZO 实时数据压缩库”, 0.28 版用户手册, 网址:<http://www.infosys.at/Staff/lux/marco/lzo.html> (1997 年 2 月), 2005 年。
- [19] 米 (meter 的缩写) 纳尔逊, “LZW 数据压缩”, 多布博士期刊, 第 14 卷, 第 10 期, 第 29-36 页, 1989 年。
- [20] 长度 P. Deutsch, “DEFLATE 压缩数据格式规范 1.3 版”, RFC 编辑器, <http://tools.ietf.org/html/rfc1951>, 1996。
- [21] X. 林, 卢, 杜力斯, 席兰和华莱士, “多级压缩:粗粒度数据重新排序以提高可压缩性”, 载于第 12 届 USENIX 文件和存储技术会议记录 (FAST' 14)。美国加州圣克拉拉:USENIX 协会, 2014 年 2 月, 第 257-271 页。
- [22] B. 布鲁姆, “带有可允许错误的散列编码中的空间/时间权衡”, 《美国计算机学会通讯》, 第 13 卷, 第 7 期, 第 422-426 页, 1970 年。
- [23] 米 (meter 的缩写) Lillibridge, K. Eshghi 和 D. Bhagwat, “提高使用内联基于块的重复数据删除的备份系统的恢复速度”, 载于第 11 届 USENIX 文件和存储技术会议记录 (FAST' 13)。美国加州圣何塞:USENIX 协会, 2013 年 2 月, 第 183-197 页。
- [24] D. 汉密尔顿, “重复数据删除-实现数据效率的方法” [http://www.snia.org/sites/default/education/tutorials/2008/spring/data-management/Hamilton-Deduplication\\_Methods\\_Data\\_Efficiency.pdf](http://www.snia.org/sites/default/education/tutorials/2008/spring/data-management/Hamilton-Deduplication_Methods_Data_Efficiency.pdf), 2008。
- [25] T. Riveria, “了解重复数据删除” [http://www.snia.org/sites/default/education/tutorials/2009/fall/data/ThomasRiveria\\_UnderstandingDeduplication\\_ATutorial\\_Understanding\\_Dedupe\\_9-15-09.pdf](http://www.snia.org/sites/default/education/tutorials/2009/fall/data/ThomasRiveria_UnderstandingDeduplication_ATutorial_Understanding_Dedupe_9-15-09.pdf), 2009。
- [26] A. Brinkmann, “重复数据删除-教程” [https://pc2.uni-paderborn.de/fileadmin/pc2/media/staffweb/Andre\\_Brinkmann/Courses/Speichersysteme\\_SS\\_2011/Deduplication\\_-\\_Eurosyst\\_Tutorial.pdf](https://pc2.uni-paderborn.de/fileadmin/pc2/media/staffweb/Andre_Brinkmann/Courses/Speichersysteme_SS_2011/Deduplication_-_Eurosyst_Tutorial.pdf), 2011。
- [27] T. Riveria 和 G. Nagle, “高级重复数据删除概念” [http://www.snia.org/sites/default/education/tutorials/2011/fall/DataProtectionManagement/ThomasRiveria\\_AdvancedDedupe\\_Concepts\\_FINAL.pdf](http://www.snia.org/sites/default/education/tutorials/2011/fall/DataProtectionManagement/ThomasRiveria_AdvancedDedupe_Concepts_FINAL.pdf), 2011。
- [28] 名词 (noun 的缩写) 曼德尔, 周, 史密斯和乌坦坦-达尼, “去神秘化重复数据删除”, 在 ACM/IFIP/USENIX 中间件' 08 会议记录中。比利时布鲁日:奥地利中心, 2008 年 12 月, 第 12-17 页。
- [29] A. F. Banu 和 C. Chandrasekar, “重复数据删除方法调查”, 《国际计算机趋势和技术杂志》, 第 3 卷, 第 3 期, 第 364-368 页, 2012 年。
- [30] 何, 李, 张, “重复数据删除技术”, 载《2010 未来信息技术与管理工程国际会议论文集》第 1 卷。中国常州:IEEE, 2010 年 8 月, 第 430-433 页。
- [31] J. Paulo 和 J. Pereira, “存储重复数据删除系统的调查和分类”, ACM 计算调查 (CSUR), 第 47 卷, 第 1 期, 第 11 页, 2014 年。
- [32] 页 (page 的缩写) Neelaveni 和 M. Vijayalakshmi, “云存储中重复数据删除的调查”, 《亚洲信息技术杂志》, 第 13 卷, 第 6 期, 第 320-330 页, 2014 年。
- [33] 页 (page 的缩写) 数据匹配:记录链接、实体解析和重复检测的概念和技术。纽约:斯普林格·弗拉格, 2012 年。
- [34] F. 瑞曼和赫歇尔, “重复检测导论”, 《数据管理综合讲座》, 第 2 卷, 第 1 期, 第 1-87 页, 2010 年。
- [35] J. 数据压缩:方法和理论。计算机科学出版社, 1988 年。
- [36] 数据压缩理论 <http://www.data-compression.com/>。
- [37] J. 盖利和阿德勒, “gzip 压缩机” <http://www.gzip.org/>, 1991。
- [38] 米 (meter 的缩写) 图像压缩基础, 标准和实践:图像压缩基础, 标准和实践。斯普林格, 2002 年, 第 1 卷。
- [39] C. 香农, “通信的数学理论”, 《移动计算与通信评论》, 第 5 卷, 第 1 期, 第 3-55 页, 2001 年。
- [40] D. A. Huffman, “一种构造最小冗余码的方法”, 《无线电工程师学会学报》, 第 40 卷, 第 9 期, 第 1098-1101 页, 1952 年。
- [41] G. 小 g. 兰登, “算术编码导论”, 《IBM 研究与发展杂志》, 第 28 卷, 第 2 期, 第 135-149 页, 1984 年。
- [42] 威廉·尼尔和克利里, “数据压缩的算术编码”, 《美国计算机学会通讯》, 第 30 卷, 第 6 期, 第 520-540 页, 1987 年。
- [43] “放气压缩” <http://zh.wikipedia.org/zh-cn/DEFLATE>, 1991。
- [44] “7zip” <http://www.7-zip.org/>。
- [45] J. Gilchrist, “使用 bzip2 的并行数据压缩”, 载于第 16 届国际并行和分布式计算与系统会议 (ICPADS' 04) 第 16 卷。美国加利福尼亚州纽波特比奇:IEEE 计算机学会出版社, 2004 年 7 月, 第 559-564 页。
- [46] 米 (meter 的缩写) 布伦斯和 d. 惠勒, “一种块排序无损数据压缩算法”, 载于《数字 SRC 研究报告》, 1994 年。
- [47] W. F. Tichy, “版本控制系统”, 软件:实践和经验, 第 15 卷, 第 7 期, 第 637-654 页, 1985 年。
- [48] J. J. Hunt, k-p. Vo, 和 W. F. Tichy, “Delta 算法:经验分析”, 《软件工程和方法论的 ACM 交易》, 第 7 卷, 第 2 期, 第 192-214 页, 1998 年。
- [49] J. 文件系统支持增量压缩硕士论文。加州大学伯克利分校电气工程和计算机科学系, 2000 年。
- [50] A. 特里吉和 p. 麦克斯, “同步算法”, 1996 年。
- [51] 名词 (noun 的缩写) 贾恩, m. 达林和 r. 特瓦里, “锥形:分层方法



- 用于消除副本同步中的冗余。“在USENIX文件和存储技术会议(FAST'05)的开幕式上。美国加州三藩市:USENIX协会,2005年3月,第281-294页。
- [52] T. Suel 和 N. Memon, “增量压缩和远程文件同步的算法”, 无损压缩手册, 2002年。
- [53] R. 彭斯和龙, “带增量压缩的高效分布式备份”, 载于《第五届并行和分布式系统输入输出研讨会论文集》。美国加利福尼亚州圣何塞:美国基督教协进会, 1997年11月, 第27-36页。
- [54] 页 (page 的缩写) Shilane, G. Wallace, M. Huang 和 W. Hsu, “使用流通知本地性的增量压缩和重复数据消除存储”, 在第四届存储和文件系统热点话题 USENIX 会议记录 (HotStorage'12) 中。美国马萨诸塞州波士顿:USENIX 协会, 2012年6月, 第201-214页。
- [55] W. 夏, 姜海峰, 田立平, “结合重复数据删除和增量压缩实现备份数据集的低开销数据减少”, 载于《IEEE 数据压缩会议论文集》(DCC'14)。美国犹他州雪鸟:IEEE 计算机学会出版社, 2014年3月, 第203-212页。
- [56] W. J. Bolosky, J. R. Douceur, D. Ely 等, “在一组现有的台式电脑上部署无服务器分布式文件系统的可行性”, 《ACM SIGMETRICS 性能评估评论》, 第28卷, 第1期, 第34-43页, 2000年。
- [57] D. Meister 和 A. Brinkmann, “备份场景中重复数据删除的多级比较”, 载于《SYSTOR 2009 会议录:以色列实验系统会议》。以色列海法:美国基督教协进会, 2009年5月, 第1-12页。
- [58] 米 (meter 的缩写) Lillibridge, K. Eshghi, D. Bhagwat 等, “稀疏索引:使用采样和局部性的大规模线内重复数据删除。”在第七届 USENIX 文件和存储技术会议记录 (FAST'09) 第9卷中。加州圣何塞:USENIX 协会, 2009年2月, 第111-123页。
- [59] C. 杜布尼奇, l. 格里兹, l. 赫尔特, m. 卡兹马尔奇克, w. 基里安, 页 (page 的缩写) 《HYDRAstor:一种可扩展的二级存储》在 USENIX 文件和存储技术会议记录 (FAST'09) 第9卷中。美国加州圣何塞:USENIX 协会, 2009年2月, 第197-210页。
- [60] D. 巴格瓦特, K. Eshghi, D. D. Long 等, “极限宁波:基于块的文件备份的可扩展并行重复数据删除”, 在 IEEE 计算机和电信系统建模、分析和仿真国际研讨会会议录 (MASCOTS'09) 中。英国伦敦:IEEE 计算机学会出版社, 2009年9月, 第1-9页。
- [61] B. Debnath, S. Sengupta, J. Li, “ChunkStash:使用闪存加速线内存储重复数据删除”, 载于2010年年度技术会议会议记录('10)。美国马萨诸塞州波士顿:USENIX 协会, 2010年6月, 第1-14页。
- [62] F. 郭和 P. Efstathopoulos, “构建高性能重复数据消除系统”, 载于2011年在举行的年度技术会议记录('11)。美国俄勒冈州波特兰市:USENIX 协会, 2011年6月, 第1-14页。
- [63] W. 夏海江, 冯德华, “筒仓:一种低内存开销、高吞吐量的基于相似局部性的近似精确重复数据删除方案”, 载于2011年年度技术会议论文集(11)。美国俄勒冈州波特兰市:USENIX 协会, 2011年6月, 第285-298页。
- [64] K. Srinivasan, T. Bisson, G. Goodson 和 K. Voruganti, “iDedup:主存储的延迟感知型线内重复数据消除”, 载于第十届 USENIX 文件和存储技术会议记录 (FAST'12)。美国加州圣何塞:USENIX 协会, 2012年2月, 第24-37页。
- [65] 动词 (verb 的缩写) 汉森, “哈希比较分析”《第九届操作系统热点研讨会论文集》(HotOS '03)。美国夏威夷州:USENIX 协会, 2003年5月, 第13-18页。
- [66] J. 布莱克, “哈希比较:一个合理的分析。”在 USENIX 年度技术会议上, 通用赛道。美国波士顿:USENIX 协会, 2006年5月, 第85-90页。
- [67] Y. 邢, 李, 戴, “对等计算:对等辅助采样重复数据删除的认识”, 2010年第十届国际对等计算会议论文集 (P2P'10)。荷兰代尔夫特:IEEE 计算机学会出版社, 2010年8月, 第1-10页。
- [68] W. 夏, 姜海峰, 田立田, 傅敏和周永安, “基于重复数据消除的快速增量压缩方法”, 性能评估, 第79卷, 第258-272页, 2014年。
- [69] D. 伊斯特莱克和琼斯, “美国安全散列算法 1 (SHA1)”, 2001年。
- [70] D. “高性能计算存储系统中的重复数据删除研究”, 载于《高性能计算、网络、存储和分析国际会议论文集》(SC'02)。美国犹他州盐湖城:IEEE 计算机学会出版社, 2012年6月, 第1-11页。
- [71] C. 里格尔和 S. G. 麦卡锡, “磁盘驱动器纠错系统的设计”, IEEE 磁学杂志, 第34卷, 第4期, 第2362-2371页, 1998年。
- [72] B. 施罗德和吉布森, “现实世界中的磁盘故障:100 万小时的 mttf 对你来说意味着什么?”在第五届 USENIX 文件和存储技术会议记录 (FAST'07) 第7卷中。美国加州圣何塞:USENIX 协会, 2007年2月, 第1-16页。
- [73] “Zfs,” <http://en.wikipedia.org/wiki/ZFS>。
- [74] I. Drago, M. Mellia, M. Munafo 等, “在 Dropbox 内部:了解个人云存储服务”, 载于《2012年互联网测量会议 ACM 会议论文集》(IMC'12)。美国马萨诸塞州波士顿:美国计算机学会协会, 2012年11月, 第481-494页。
- [75] E. “双峰内容定义的备份流分块”, 载于《第七届 USENIX 文件和存储技术会议论文集》(FAST'10)。美国加州圣何塞:USENIX 协会, 2010年2月, 第1-14页。
- [76] W. 夏, 江海峰, 田立田, 傅敏敏, 王泽军, “重复数据删除:利用重复数据删除系统中的并行性”, 载于第七届网络、体系结构和存储国际会议录 (NAS'12)。中国厦门:IEEE 计算机学会出版社, 2012年6月, 第338-347页。
- [77] 米 (meter 的缩写) 随机多项式指纹。计算技术研究中心。艾肯计算实验室, 大学, 1981年。
- [78] K. 基于内容的分块算法的分析和改进框架。惠普实验室, 帕洛阿尔托, 2005年。
- [79] D. 博巴琼, 杜布尼奇和贾根纳坦, “指纹差异:存储系统中改进的重复消除”《海量存储系统与技术学报》(MSST 06年版)。美国马里兰州大学公园:IEEE 计算机学会出版社, 2006年5月, 第1-5页。
- [80] D. Teodosiu, N. Bjorner, Y. Gurevich, M. Manasse 和 J. Porkka, “使用远程差分压缩优化有限带宽网络上的文件复制”, 微软研究 TR-2006-157, 2006。
- [81] B. 阿加尔瓦尔, 阿凯拉, 阿南德等, “EndRE:企业端系统冗余消除服务”, 载于《第七届 USENIX 网络系统设计与实现会议论文集》(NSDI'10)。美国加州圣何塞:USENIX 协会, 2010年4月, 第14-28页。
- [82] Y. 张, 蒋海平, 冯德伟, 夏文伟, 傅明伟, 黄福飞, 和 Y. 周, “AE:一种非对称极值内容定义的分块算法, 用于快速且带宽高效的数据重复数据消除”, 载于《IEEE INFOCOM 2015 会议录》。香港:IEEE 计算机学会出版社, 2015年4月, 第1-9页。
- [83] C. 于, 张春生, 毛永和, 李福林, “基于熵的内容定义分块——理论与实现”, 载于第31届大容量存储系统与技术研讨会论文集('15)。美国加州圣克拉拉:IEEE, 2015年6月, 第1-12页。
- [84] B. 罗曼滑雪。Heldt, W. Kilian, K. Lichota 和 C. Dubnicki, “锚驱动的子块重复数据删除”, 载于第四届国际系统和存储年会论文集 (SYS-TOR'11)。以色列海法:美国基督教协进会, 2011年5月, 第1-13页。
- [85] G. 陆, 金, 杜德华, “基于频率的分块去重技术”, 2010年 IEEE 计算机与电信系统建模、分析与仿真国际研讨会论文集 (MASCOTS'10)。美国佛罗里达州迈阿密海滩:IEEE 计算机学会出版社, 2010年8月, 第287-296页。
- [86] B. 周和温, “基于滞后重分块的元数据利用磁盘映像的重复数据删除”, 载于第42届国际并行处理会议录('13)。法国里昂:IEEE 计算机学会出版社, 2013年10月, 第389-398页。
- [87] A. 布罗德, “拉宾指纹法的一些应用”  
序列二:通信、安全和计算机中的方法

《科学》，第1-10页，1993年。

[88]崔永安，赖振华，王晓军，戴恩华，苗振华，“快手：提高移动云存储的同步效率服务”，载于《第21届国际年会论文集》关于移动计算和网络(MobiCom'15)。法国巴黎：ACM，2015年9月，第592-603页。

89 a. Anand, C. Muthukrishnan, A. Akella and R. Ramjee, "Redun-网络流量中的舞蹈：发现与启示 第十一届国际计量与测量联合会议纪要 计算机系统建模(SIGMETRICS-性能2009)。美国华盛顿州西雅图市：美国计算机学会协会，2009年6月，第37-48页。

90 比约纳、布拉斯和古列维奇，“内容依赖差分压缩的分块，本地最大 approach”，“计算机与系统科学杂志”，第76卷，第3期，第154-203页，2010年。

[91]博巴荣格、贾根纳坦和杜布尼奇，“改进存储系统中的重复消除”的文章 存储(TOS)，第2卷，第4期，第424-448页，2006年。

[92]赵敏，尹德元，“高效的重复数据删除技术对于现代备份操作”，“计算机上的IEEE事务”，第60卷，第6期，第824-840页，2011年。

[93]刘振华，薛玉英，朱婷等，“一种新的优化方法提高重复数据删除存储系统的性能 第十五届国际并行与分布式会议纪要 系统(ICPADS'09)。中国深圳：IEEE计算机学会出版社，2009年12月，第228-235页。

[94]马骏，赵斌，王庚，刘骏，“自适应流水线重复数据删除”，载于第28届IEEE质量研讨会论文集 存储系统和技术(MSST'12)。加利福尼亚州太平洋小树林，美国：IEEE计算机学会出版社，2012年4月，第1-6页。

95s. Al-kismwany, A. Gharaibeh, E. Santos-Neto等人，"Storegpu：利用图形处理单元加速分布式存储系统”，载于第17届国际研讨会论文集 关于高性能分布式计算(HPDC'08)。波士顿，美国文学硕士：美国文学硕士协会，2008年6月，第165-174页。

[96]p. Bhatita, R. Rodrigues and a. 维尔马，“碎纸机：图形处理器加速增量存储和计算”中 第十届USENIX文件和存储技术会议(FAST'12)。美国加利福尼亚州圣何塞：USENIX协会，2月2012年，第1-15页。

[97]金正云、朴光熙和朴光熙，“幽灵：图形处理器-卸载主存储的高性能存储I/O重复数据删除系统”，载于2012年国际研讨会论文集 多内核和多内核的编程模型和应用。ACM，2012年，第17-26页。

[98]林、杜格利什、李、李、瑞西、斯梅尔顿和G.华莱士，“元数据被认为是有害的。。。去重复数据删除”，在第七届USENIX存储和文件系统热点研讨会上 (HotStorage'15)。加利福尼亚州圣克拉拉：USENIX协会，7月2015。

[99] J. Bowling，“开放重复数据删除：开源重复数据删除测试”，《Linux杂志》，第2013卷，第228期，第2页，2013年。

[100]夏文伟，江海红，冯德华，田立平，“加速数据去度-利用流水线技术和多核并行技术实现应用或众核处理器”，载于《第十届USENIX会议论文集 文件和存储技术会议(FAST'12海报)。疗养院 美国加州圣何塞：USENIX协会，2012年2月，第1-2页。

[101] A. Gharaibeh, s. Al-kishwany, S. Gopalakrishnan等人，“一个图形处理器加速存储系统”，载于第19届美国计算机学会会员 高性能分布式计算国际研讨会(HPDC'10)。美国伊利诺伊州芝加哥：美国计算机学会协会，2010年6月，第167-178页。

[102]夏文伟，江海红，冯德华，“相似性与局部性”基于索引的高性能重复数据删除”，IEEE《计算机学报》，第64卷，第4期，第1162-1176页，2015年。

[103]傅敏德，冯玉华，何晓军，陈泽军，夏文伟，张永和Y.谭，“中重复数据删除性能的设计权衡备份工作负载”，载于第13届USENIX会议记录 关于文件和存储技术(FAST'15)。美国加利福尼亚州圣克拉拉市：USENIX协会，2015年2月，第331-344页。

[104]魏俊杰，姜海红，周k等，“MAD2：一种可扩展的高-网络备份的吞吐量精确重复数据删除方法服务”，载于2010年IEEE第26届质量研讨会论文集 存储系统和技术(MSST'10)。倾斜村，美国内华达州：IEEE计算机学会出版社，2010年5月，第1-14页。

第39届国际并行处理会议

(ICPP'10)。美国加州圣地亚哥：IEEE计算机学会出版社，2010年9月，第614-623页。

[106] D. “块局部缓存对于重复数据消除”，载于第六届国际会议论文集 系统和存储会议(Systor'13)。以色列海法：巴勒斯坦基督教协进会协会，2013年6月，第1-12页。

[107] 长度阿罗诺维奇，阿瑟，巴赫马特等人，“一个基于相似性的重复数据删除系统”，发表在《SYSTOR学报》上 2009年：以色列实验系统会议。以色列海法：ACM协会，2009年5月，第1-12页。

[108] D. 梅斯特和布林克曼，“重复数据删除1：改进重复数据删除-固态硬盘(ssd)的吞吐量”，载于 IEEE第26届大容量存储系统和技术研讨会(MSST'10)。美国内华达州倾斜村：IEEE计算机社会出版社，2010年5月，第1-6页。

[109] G. 吕燕南和杜德华，“布隆商店：布隆过滤器用于数据索引的基于内存的高效键值存储闪存上的重复数据删除”，载于IEEE第28届研讨会论文集 大容量存储系统和技术(MSST'12)。太平洋小树林，美国加州：IEEE计算机学会出版社，2012年4月，第1-11页。

[110] T. 杨，江海峰，等，“DEBAR：一个可扩展的高用于备份和恢复的高性能重复数据删除存储系统存档”，载于2010年IEEE国际研讨会论文集 关于并行和分布式处理。美国亚特兰大：IEEE计算机学会出版社，2010年4月，第1-12页。

[111] W. 董，李，等，“可伸缩数据中的权衡重复数据删除集群的路由。《第九届会议记录》 USENIX文件和存储技术会议(FAST'11)。疗养院 美国加州圣何塞：USENIX协会，2011年2月，第15-29页。

[112] Y. 付，蒋海辉，肖恩华，“一个可扩展的内联集群

大数据保护的应用框架”，载于 美国计算机学会/IFIP/USENIX中间件会议。魁北克蒙特利尔，加拿大：斯普林格，2012年12月，第354-373页。

[113] J. 凯撒、梅斯特、布林克曼和埃弗特，“设计一个精确的重复数据删除集群”，载于2012年IEEE第28届会议记录 大容量存储系统和技术研讨会(MSST12)。美国加利福尼亚州太平洋格罗夫：IEEE计算机学会出版社，4月2012年，第1-12页。

[114] D. 弗雷、克雷马克和克劳达斯，“概率性的”基于集群的存储系统的应用”，载于 第三届中国计算机学会云计算研讨会(SOCC'12)。圣何塞，美国加州：美国计算机学会协会，2012年10月，第1-12页。

[115] A. 论文件的相似性和包容性 “序列的压缩和复杂性论文集”(SEQUENCES '97)。美国DC华府：IEEE，1997年6月，页。21-29。

[116] B. 《闪存：高吞吐量》持久的键值存储，”VLDB捐赠基金，第3卷，第1-2期，第1414-1425页，2010年。

[117] ——“存储空间不足：内存空间不足的键值存储基于闪存的存储”，载于《2011年ACM SIGMOD会议录》

国际数据管理会议。

ACM，2011年，第25-36页。

[118] R. 《布谷鸟散列法》，《算法杂志》，第51卷，第2期，第122-144页，2004年。

[119] A. Broder，“识别和过滤近似重复的文档”组合模式匹配学报。蒙特利尔，

加拿大：斯普林格，2000年6月，第1-10页。

[120] 页(page)的缩写)库尔卡尼、杜利斯、拉沃伊和特雷西 大量文件中的舞蹈消除”，载于《会议记录》

2004年USENIX年度技术会议(2004年USENIX)。

美国马萨诸塞州波士顿：USENIX协会，2012年6月，第1-14页。

[121] D. 古普塔，李少光，m. 弗莱伯等，“差异引擎：利用虚拟机中的内存冗余 第八届操作系统设计与实现研讨会(OSDI'08)。美国加州圣地亚哥：USENIX协会，12月-2008年12月，第309-322页。

[122] D. 《Zdelta：一个有效的delta》压缩工具，“技术报告，计算机和 理工大学信息科学，2002年。

[123] 单位曼伯，“在大型文件系统中查找相似的文件”，在《USENIX冬季》，第94卷。美国加利福尼亚州旧金山市：USENIX协会，1994年1月，第1-10页。

[124] F. 特定应用的增量编码

[105]谭玉英, 蒋海红, 冯德明等, “语义感知的多元语义模型”  
云备份的分层源重复数据消除框架”, 在中

通过相似性检测”, 载于《USENIX 年刊》  
普通赛道技术会议。美国德克萨斯州圣安东尼奥:

USENIX 协会, 2003 年 6 月, 第 113-126 页。

[125] 刘立友、波拉克和龙, “深藏: 档案馆存储系统架构”, 载于《21 世纪记录》

国际数据工程会议 (05 年 ICDE)。东京,

日本: IEEE 计算机学会出版社, 2005 年 4 月, 第 804-815 页。

[126] 杨, 任, “智能现金: 智能耦合阵列固态硬盘和硬盘”, 载于第 17 届 IEEE 国际会议录高性能计算机体系结构研讨会 (HPCA'11)。

美国德克萨斯州圣安东尼奥: IEEE 计算机学会出版社, 2 月 2011 年, 第 278-289 页。

[127] 夏文伟, 姜海峰, 田立平, “重复数据删除

数据的感知相似性检测和消除方案  
低管理费用的减少”, IEEE 计算事务  
ers, 第 1-14 页, 2016 年。

[128] 南永铨、朴志和杜, “保证要求阅读——  
备份数据集的重复数据消除存储性能,”

在 IEEE 第 20 届国际建模研讨会会议录中,  
计算机和电信系统的分析与模拟  
(MASCOTS'12)。美国 DC 华盛顿州: IEEE 计算机学会  
出版社, 2012 年 8 月, 第 201-208 页。

[129] 傅敏德, 冯玉华, 何晓军, 陈泽军, 夏文伟, 黄福荣,  
和刘, “加速恢复和垃圾收集  
通过利用历史  
信息”, 载于 2014 年 USENIX 年度技术会议记录  
会议 (USENIX 空管 14)。美国宾夕法尼亚州费城: USENIX  
协会, 2014 年 6 月, 第 181-192 页。

[130] 毛, 江海平, 吴树声, 田立平, “POD: 性能  
中主存储系统的定向输入/输出重复数据消除  
云”, 载于 2014 年 IEEE 第 28 届国际平行和  
分布式处理研讨会 (IPDPS'14)。美国亚利桑那州凤凰城:  
IEEE, 2014 年 5 月, 第 767-776 页。

[131] M. 卡兹马尔奇克、m. 巴尔钦斯基、w. 基里安和 c. 杜布尼奇,  
“减少由在线重复数据消除造成的数据碎片的影响—  
第五届国际系统年会论文集

和存储会议 (SYSTOR'12)。以色列海法: ACM Associ-  
《教育》, 2012 年 6 月, 第 1-12 页。

[132] 吴春华和李炳平, “反向重复数据删除: 反向重复数据删除  
存储系统针对读取最新备份进行了优化  
第四届亚太系统研讨会综述。澳大利亚竞争管理委员会, 2013 年,  
页 (page 的缩写) 15。

[133] 谭玉英, 蒋海红, 冯德伟等, “CABdedupe: 一个基于因果关系的研  
究”  
云备份服务的重复数据消除性能助推器,  
2011 年 IEEE 国际并行与分布式会议论文集  
加工研讨会 (IPDPS'11)。美国阿拉斯加安克雷奇:  
IEEE 计算机学会出版社, 2011 年 5 月, 第 1266-1277 页。

[134] 毛斌, 蒋海红, 吴素珊等, “合成孔径雷达: 固态硬盘辅助恢复  
中基于重复数据消除的存储系统的优化  
云”, 载于 2012 年第七届 IEEE 国际会议论文集  
网络、体系结构和存储 (NAS'12)。中国厦门:  
IEEE 计算机学会出版社, 2012 年 6 月, 第 328-337 页。

[135] 赖瑞华, 冯德华, 夏文伟, 傅敏英, 杨玉英, “一近——  
提高还原性能的精确碎片整理方案  
对于云备份系统”, 载于《第 14 届国际会议论文集》  
并行处理算法和体系结构会议  
(ICA3PP'14)。斯普林格, 2014 年, 第 457-471 页。

[136] 南永南, 陆国栋, 朴永国, 肖文伟, 杜德海, “大块  
碎片级别: 读取性能的有效指标  
重复数据消除存储的退化”, 载于《2011 年会议录》  
IEEE 第十三届高性能计算国际会议  
和通信 (HPCC'11)。加拿大班夫: IEEE 计算机

社会出版社, 2011 年 9 月, 第 581-586 页。

[137] 李彦宏, 徐明辉, 吴春华, 李炳平, “高效混合动力直列和  
备份存储的线外重复数据消除”, “ACM 事务  
关于存储 (TOS), 第 10 卷, 第 2 期, 第 2-21 页, 2014 年。

[138] 傅敏德, 冯玉华, 何晓军, 陈子珍, 刘建华, 夏文伟, 黄福荣,  
和刘, “减少在线重复数据消除的碎片  
通过利用备份历史和缓存知识来备份存储  
edge”, “并行和分布式系统的 IEEE 事务”, 第 27 卷,  
第 3 期, 第 855-868 页, 2016 年。

[139] L. Belady, “一个虚拟的替换算法的研究——  
存储计算机”, IBM 系统杂志, 第 5 卷, 第 2 期, 第 78-101 页,  
1966。

[140] 毛, 江海平, 吴树声, 傅友年, 田立军, “阅读表演

增量备份的操作和垃圾收集引擎”, 在

第六届国际系统与存储会议录  
(SYSTOR'13)。以色列海法: 美国计算机学会, 2013 年 6 月, 第 1-  
12 页。

[142] 页 (page 的缩写) Strzelczak, E. Adamczyk, U. Herman-  
Izycka, J. Sakowicz,  
长度 Slusarczyk, J. Wrona 和 C. Dubnicki, “同时删除  
在分布式内容可寻址存储系统中  
重复数据删除。 “在第 11 届 USENIX 会议记录上  
文件和存储技术 (FAST'13)。美国加利福尼亚州圣何塞市:  
USENIX 协会, 2013 年 2 月, 第 161-174 页。

[143] 动词 (verb 的缩写) 塔拉斯夫、贾恩、恩宁、曼达尔、帕拉尼萨  
米、  
页 (page 的缩写) 夏兰, s. 特雷汉和 e. 扎多克, “数据映射器:  
设备映射器  
重复数据消除的目标”, 载于《渥太华 Linux 学报》  
研讨会, 加拿大渥太华, 2014 年 7 月, 第 83-87 页。

[144] F. 博特利霍、夏兰、加格和许, “记忆效率  
已消除重复数据的存储系统的清理”中  
第 11 届 USENIX 文件和存储技术会议  
(FAST'13)。美国加利福尼亚州圣何塞: USENIX 协会, 2 月  
2013 年, 第 81-94 页。

[145] F. 博特利霍、拉塞达、梅内塞斯和日维亚尼,  
“最小完美散列法: 一种有竞争力的索引方法  
内部记忆”, 《信息科学》, 第 181 卷, 第 13 期, 第 2608 页 -  
2625, 2011。

[146] 米 (meter 的缩写) 穆拉扎尼、施里特维瑟、莱什纳、胡贝尔和  
E. Weippl, “地平线上的乌云: 使用云存储作为  
攻击向量和在线松弛空间。《第 20 届会议记录》  
USENIX 安全研讨会。美国加利福尼亚州三藩市: USENIX  
协会, 2011 年 8 月, 第 1-11 页。

[147] D. 哈尼克, 平卡斯和舒尔曼-皮莱格, “边频道在  
云服务: 云存储中的重复数据消除, “IEEE 安全 &  
隐私, 第 8 卷, 第 6 期, 第 40-47 页, 2010 年。

[148] 南哈列维, 哈尼克, 平卡斯和舒尔曼-皮莱格, “的证明  
远程存储系统的所有权”, 载于《第 18 届会议记录》  
计算机和通信安全会议。  
美国伊利诺伊州芝加哥: 美国计算机学会协会, 2011 年 10 月, 第  
491 页 -  
500。

[149] T. 用于备份文件的系统  
计算机网络多个节点上的磁盘卷”, 7 月 7 日  
1998 年, 美国专利 5, 778, 395。

[150] J. 杜塞尔、阿迪亚、博洛斯基等人, “回收空间  
无服务器分布式文件系统重复文件  
在第 22 届国际分布式计算会议记录中

计算系统 (ICDCS'02)。奥地利维也纳: IEEE 计算机  
社会出版社, 2002 年 7 月, 第 617-624 页。

[151] 长度考克斯、默里和诺贝尔, “仿作: 制作”  
备份便宜又容易”, ACM SIGOPS 操作系统评论,  
第 36 卷, 国际标准化组织编号, 第 285-298 页, 2002 年。

[152] 米 (meter 的缩写) 《安全》  
重复数据删除”, 载于第四届 ACM 国际会议录  
存储安全性和可生存性研讨会。亚历山大,  
美国弗吉尼亚州: 美国计算机学会协会, 2008 年 10 月, 第 1-10  
页。

[153] 页 (page 的缩写) 安德森和张, “快速安全的笔记本电脑备份  
加密的重复数据删除”, 载于第 23 届国际  
大型安装系统管理会议: 策略,  
工具和技术 (LISA'10)。马萨诸塞州波士顿: USENIX  
协会, 2010 年 12 月, 第 195-206 页。

[154] J. 李、李俊杰、李炳平、陈晓明、李明辉和楼伟, “安全重复数  
据删除—  
阳离子具有高效可靠的聚合密钥管理, ”  
IEEE 并行和分布式系统事务, 第 1-11 页,  
2013。

[155] 米 (meter 的缩写) 贝拉雷, s. 基尔维迪和 t. 里斯坦部  
分, “信息锁定  
加密和安全的重复数据删除”, 载于《进步学报》  
在密码学 - EUROCRYPT 2013 中。雅典: 斯普林格, 2013 年 5 月,  
第 296-312 页。

[156] —— “双重: 针对已消除重复数据的存储的服务器辅助加密”  
年龄”, 载于第 22 届 USENIX 安全研讨会论文集。  
美国 DC 华府: USENIX 协会, 2013 年 8 月, 页  
1-16。

[157] O. 海伦, c. 诺依曼, l. 蒙塔尔沃和 s. 德弗兰斯, “我—  
证明了对云存储侧信道攻击的抵抗力  
服务”, 载于《第五届新技术国际会议论文集》  
技术、移动性和安全性 (NTMS'12)。土耳其伊斯坦布尔:  
IEEE 计算机学会出版社, 2012 年 5 月, 第 1-5 页。

[158] R.C. Merkle, “经认证的数字签名”, 载于《进展》

- 中基于重复数据消除的存储系统的优化
- 云, “存储上的 ACM 事务 (TOS)”, 第 10 卷, 第 2 期, 第 6 页, 2014.
- [141] 西姆哈、卢和丘埃, “一种可扩展的重复数据消除方法”
- [159] *Cryptology* / CRYPTO ' 89 会议录。斯普林格, 1989 年, 第 218-238 页。
- R. 迪·皮埃特罗和阿·索尼奥蒂, “提高效率和安全性在重复数据消除的所有权证明”中
- 第七届美国计算机学会信息、计算机和通信研讨会



- 安全(ASIACCS'12)。韩国首尔:美国计算机学会协会, 5月2012年, 第81-82页。
- [160]郑, 徐, “安全高效的存储证明重复数据消除”, 载于第二届ACM数据会议记录以及应用安全和隐私。美国德克萨斯州圣安东尼奥:ACM协会, 2012年2月, 第1-12页。
- [161]徐军, 张华东, 周军, “弱泄漏-弹性客户端-云存储中加密数据的并行重复数据消除, 请参见第八届亚洲计算机学会信息专题讨论会论文集, 计算机和通信安全(ASIACCS'13)。杭州, 中国:美国计算机学会, 2013年5月, 第195-206页。
- [162]周永年, 冯德伟, 夏文伟, 傅明福, 黄福福, 张永年, 李春华, SecDep:一种用户感知的高效细粒度安全重复数据消除具有多级密钥管理的加密方案”第31届大容量存储系统和技术研讨会(MSST'15)。美国加州圣克拉拉:IEEE, 2015年6月, 第1-12页。
- [163]李明钦, 李炳平, “CDStore:走向可靠, 安全, 以及通过聚合分散实现的经济高效的云存储,” 中2015年USENIX年度技术会议论文集会议。美国加利福尼亚州圣克拉拉:USENIX协会, 7月2015年, 第111-124页。
- [164]李明扬, 秦振中, 李炳平, 李俊杰, “收敛扩散:向云环境中的存储高效安全”, 载于第六届USENIX存储和文件系统热点研讨会(HotStorage'14)。美国宾夕法尼亚州费城:USENIX协会, 2014年6月, 第1-5页。
- [165]巴格瓦特、波拉克、朗、施瓦茨、米勒和J.-f. Paris, “以最少的冗余提供高可靠性档案存储系统”, 载于第14届国际电气和电子工程师协会会议记录计算机建模、分析和模拟全国研讨会 and 电信系统(MASCOTS'06)。加利福尼亚州蒙特雷市, 美国:IEEE计算机学会出版社, 2006年9月, 第413页-421。
- [166]李, 李桥和, “可靠性分析经过重复数据消除和擦除编码的存储《绩效评估评论》, 第38卷, 第3期, 第4-9页, 2011年。
- [167]刘振华, 顾永年, 孙立军, 颜伯和王德生, “高为大规模重复数据消除归档存储提供可靠性年龄系统”, 载于第23届老龄问题国际会议记录超级计算。美国纽约州约克镇高地:美国计算机学会协会, 2009年6月, 第370-379页。
- [168]李k, “新兴技术:DD200恢复器”<http://storageconference.us/2004/Presentations/Panel/KaiLi.pdf> ; 2004年4月。
- [169] E. W. Rozier, W. H. Sanders, P. Zhou, N. Mandagere, S. M. Ut-坦尚达尼和雅库舍夫, “容错建模重复数据删除的后果”, 载于《第30届IEEE会议记录可靠分布式系统研讨会(SRDS'11)。马德里, 西班牙:IEEE, 2011年10月, 第75-84页。
- [170]康斯坦丁内斯库和卢, “数据的快速估算”面向大型存储系统的压缩和重复数据消除第一届国际数据压缩会议纪要, 通信和处理(CCP'11)。意大利帕利努罗:IEEE, 2011年6月, 第98-102页。
- [171]汤旺森、普查、安德森和卡明斯基, “利用数据重组的系统的有效相似性估计邓丹西”, 载于《IEEE信息通信会议记录》。加利福尼亚州圣地亚哥市, 美国:IEEE, 2010年3月, 第1-9页。
- [172]谢, 康迪特和谢, “估计重复基于内容的采样”, 载于《2013年USENIX会议记录年度技术会议会议(USENIX'13)。圣何塞, 美国加州:USENIX协会, 2013年6月, 第181-186页。
- [173] D. Harnik, O. Margalit, D. Naor, D. Sotnikov and G. Vernik, “Es-大型数据集中重复数据消除率的估算”, 载于《会议记录》IEEE第28届大容量存储系统和技术研讨会(MSST'12)。美国加州太平洋小树林:IEEE, 2012年5月, 第1-11页。
- [174]梅斯特、布林克曼和苏, “文件配方压缩在重复数据删除系统中”, 载于第11届USENIX会议记录文件和存储技术会议(FAST'13)。圣何塞, 美国加州:USENIX协会, 2013年2月, 第175-182页。
- [175]卡蒂亚尔和魏斯曼, “视频:应用感知视频重复数据消除框架”, 载于第三届会议记录存储和文件系统热点问题USENIX会议(热点-存储'11)。美国俄勒冈州波特兰市:USENIX协会, 2011年6月, 第1-5页。
- 机器视觉会议(BMVC'14)。英国诺丁汉:英国人机器视觉协会(BMVA), 2014年9月, 第1-12页。
- [177] Y. 华春莹, 姜海峰, “FAST:近实时可搜索云的数据分析”, 载于《国际会议》高性能计算、网络和存储会议和分析(SC'14)。IEEE出版社, 2014年, 第754-765页。
- [178] 南德瓦卡, 苏比亚, 声达拉扬, 威尔逊, 斯托托, K. 乌达亚山卡、沃鲁甘蒂和邵, “储存视频存储库的虚拟化机会和分析”中第七届USENIX存储和文件系统热点研讨会(HotStorage'15)。美国加利福尼亚州圣克拉拉:USENIX协会, 2015年7月。
- [179] R. 柯勒和朗加斯瓦米, “输入/输出重复数据消除:利用con-帐篷相似性, 以提高输入/输出性能关于存储(TOS), 第6卷, 第3期, 第13页, 2010年。
- [180] 米(meter的缩写))萨维奇和沃克尔, “积云:文件系统云备份, “存储上的ACM事务(TOS), 第5卷, 第4期, 第14页, 2009年。
- [181] C. A. Waldspurger, “VMware中的内存资源管理ESX服务器, “ACM SIGOPS操作系统评论”, 第36卷, 国际标准化组织, 第181-194页, 2002年。
- [182] A. T. Clements, I. Ahmad, M. Vilayannur, J. Li等人, “分散的存储区域网络集群文件系统中的标准化重复数据消除”, 载于2009年USENIX年度技术会议(USENIX 09年)。疗养院美国加州迭戈:USENIX协会, 2009年6月, 第1-14页。
- [183] 名词(noun的缩写)斯普林和韦瑟拉尔, “一种独立于协议的技术——用于消除冗余网络流量的nique,” ACM SIG-通信计算机通信评论, 第30卷, 第4期, 第87-95页, 2000。
- [184] A. 阿南德、维·塞卡尔和阿凯拉, “智能:一种架构对于协调的全网络冗余消除, “ACM SIGCOMM计算机通信评论, 第39卷, 第4期, 共10页87-98, 2009。
- [185] F. 陈, 罗, 张, “CAFTL:一个内容感知的flash转换层提高了闪存的寿命固态硬盘。“在第九届USENIX会议记录中关于文件和存储技术(FAST'11)。美国加利福尼亚州圣何塞市:USENIX协会, 2011年2月, 第1-14页。
- [186] C. 李、夏兰、杜力斯、辛、斯梅尔顿和沃尔花边, “硝基:一个容量优化的主存储固态硬盘缓存年龄”, 载于2014年USENIX问题USENIX会议记录年度技术会议(USENIX'14)。宾夕法尼亚州费城, 美国:USENIX协会, 2014年6月, 第501-512页。
- [187] 南雷亚、r. 考克斯和a. 劳心劳力, “快速、廉价的内容——基础中的寻址存储”, 载于2008年USENIX会议记录年度技术会议(08年USENIX)。美国加利福尼亚州圣克拉拉市:USENIX协会, 2008年6月, 第143-156页。
- [188] “打开”<http://www.opendedup.org/>
- [189] 页(page的缩写)Koutoupis, “用linux进行重复数据消除”, 《Linux杂志》, 第一卷。2011年, 第207期, 第7页, 2011年。
- [190] I. Drago, E. Bocchi, M. Mellia, H. Slatman 和 A. Pras, “法官席-标记个人云存储”, 在2013年会议记录中互联网测量会议(13年国际计量会议)会议。巴塞罗那, 西班牙:ACM协会, 2013年10月, 第205-212页。
- [191] K. 米勒、弗朗茨、里廷豪斯、希伦布兰德和贝尔“XLH:更有效的内存重复数据消除扫描仪通过跨层暗示”, 载于《2013年USENIX论文集》年度技术会议(USENIX'13)。美国加利福尼亚州圣何塞市:USENIX协会, 2013年6月, 第279-290页。
- [192] K. 金和米勒, “重复数据删除的有效性在虚拟机磁盘映像上”, 在SYSTOR'09会议记录中:以色列实验系统会议。以色列海法:巴勒斯坦基督教协进会协会, 2009年5月, 第1-14页。
- [193] J. 任、杨, “一种新的缓冲区缓存设计开发时间和内容位置”, 载于《2010年电气和电子工程师协会学报》第30届分布式计算系统国际会议(ICDCS'10)。意大利热那亚:IEEE计算机学会出版社, 6月2010年, 第273-282页。
- [194] C.-吴惠兰、马明敏、黄天佑、李炳生及吕俊杰, “现场直播开放环境中虚拟机映像的重复数据删除存储来源云”, 载于第12届国际中间件会议记录会议。葡萄牙里斯本:国际信息联合会信息处理, 2011年12月, 第80-99页。
- [195] X. 张, 霍, 马骏等, “利用重复数据删除技术加速实时虚拟机迁移”, 载于《2010年会议记录》

[176]佩拉和弗雷姆,“云尺度图像压缩  
通过内容重复数据删除”,载于《英国会议录》

*IEEE 集群计算国际会议 (CLUSTER' 10)*。  
希腊克里特岛赫拉克利翁: IEEE 计算机学会出版社, 赛特-

- 2010年12月,第88-96页。
- [196]单位“虚拟机的实时集群迁移”,《第20届高性能分布式计算国际研讨会论文集》(HPDC'11)。美国加州圣何塞:美国计算机学会协会,2011年6月,第135-146页。
- [197]H. 普查、安德森和卡明斯基,“利用文件手印的多源下载特性”,载于《第四届USENIX网络系统设计与实现会议论文集》(NSDI'07年)。美国马萨诸塞州剑桥:USENIX协会,2007年4月,第1-14页。
- [198]南 Sanadhya, R. Sivakumar, k-h. Kim, P. Congdon, S. Lakshmanan 和 J. P. Singh,“非对称缓存:改进的移动设备网络重复数据删除”,载于第18届移动计算和网络国际年会议论文集(MobiCom'12)。土耳其伊斯坦布尔:ACM协会,2012年8月,第161-172页。
- [199]Y. 华,刘 x,冯 d,“面向存储感知型SDN的智能网络内重复数据删除”,载于ACM SIGCOMM 2013年SIGCOMM会议录。中国香港:ACM协会,2013年8月,第509-510页。
- [200]Y. 华和刘,“为航空电子应用调度具有延迟感知重复数据删除的异构流”,IEEE并行和分布式系统事务,第23卷,第9期,第1790-1802页,2012。
- [201]A. 古普塔、皮索卡尔、乌尔高卡尔和西瓦苏布拉曼,“利用价值局部性优化基于NAND闪存固态硬盘。”在第九届USENIX文件和存储技术会议记录(FAST'11)中。美国加州圣何塞:USENIX协会,2011年2月,第91-103页。
- [202]Y. 唐和杨,“通用计算的安全重复数据删除”,《2015年年度技术会议论文集》,'15。美国加州圣克拉拉:USENIX协会,2015年7月,第319-331页。
- [203]“惠普:消除传统备份和归档的界限,”<http://www8.hp.com/us/en/products/data-storage/storage-backup-archive.html>,2014年4月。
- [204]HYDRASstor - 横向扩展网络存储平台<http://www.necam.com/hydrastor/>,2014年4月。
- [205]“Commvault Simpana 软件”<http://www.commvault.com/simpana-software>。
- [206]“根据IDC的数据,2013年第四季度,全球专用备份设备(PBBA)市场的收入同比增长9.7%。”<http://www.idc.com/getdoc.jsp?containerId=prUS24762914>,2014年3月。
- [207]DDBoost是“标准”吗<http://www.emc.com/data-protection/data-domain/data-domain-boost.htm>,2014年4月。
- [208]《Oracle数据库备份和恢复用户指南》[http://docs.oracle.com/cd/E11882\\_01/backup.112/e10642/rcmquick.htm#BRADV89346](http://docs.oracle.com/cd/E11882_01/backup.112/e10642/rcmquick.htm#BRADV89346)。
- [209]“赛门铁克超越虚拟磁带”<http://www.enterprisestorageforum.com/technology/news/article.php/3643846/Symantec-Looks-Beyond-Virtual-Tape.htm>,2006年11月。
- [210]“Avamar 重复数据消除备份软件 and 系统”<http://www.emc.com/domains/avamar/index.htm>。
- [211]J. 刘,柴,秦,肖,“可编程逻辑控制器高速缓存:基于重复数据消除的主存储的耐用固态硬盘高速缓存”,第30届IEEE大容量存储系统与技术研讨会论文集('14)。美国加州圣克拉拉:IEEE计算机学会出版社,2014年6月,第1-6页。
- [212]A. Wildani, E. L. Miller 和 O. Rodeh,“手:一个试探性安排的非备份在线重复数据删除系统”,在IEEE第29届国际数据工程会议(ICDE'13)的会议记录中。澳大利亚布里斯班:IEEE,2013年4月,第446-457页。
- [213]动词 (verb 的缩写) Tarasov, A. Mudrankit, W. Buik, P. Shilane, G. Kuenning 和 E. Zadok,“生成用于重复数据删除分析的现实数据集”,载于《2012年USENIX年度技术会议论文集》(USENIX'12)。美国马萨诸塞州波士顿:USENIX协会,2012年6月,第24-34页。
- [214]J. 保罗·李斯、佩雷拉和索萨,“去重复存储系统的基准”,载于《走向有意义的互联网系统:OTM 2012》。斯普林格,2012年,第584-601页。
- [215]R. 细川玉子-蒂涅多、哈尼克、纳尔、索特尼科夫、托莱多和扎克,“SDGen:在存储基准中模拟内容生成的数据集”,载于第13届USENIX文件和存储技术会议记录(FAST'15)。USENIX协会,2015年,第317-330页。

- [216]赛门铁克开放式存储<http://www.symantec.com/page.jsp?id=openstorage>,2006年11月。
- [217]B. 科恩,“BitTorrent 协议规范”<http://www.bittorrent.com/>,2008。
- [218]Y. 唐,李炳平,吕俊杰,帕尔曼,“具有访问控制和保证删除的安全覆盖云存储”,IEEE可靠和安全计算事务,第9卷,第6期,第903-916页,2012年。
- [219]A. 康斯坦丁内斯库,卢,劳雷,夏尔马,页 (page 的缩写)《重复数据删除:磁带系统的重复数据删除》,载于IEEE第30届大容量存储系统与技术研讨会论文集(MSST'14)。美国加州圣克拉拉:IEEE,2014年6月,第1-11页。

## 附录一

一些经常出现的缩写和术语出现在表中10。

表10  
首字母缩略词和术语列表。

| 条款                         | 说明                                       |
|----------------------------|--|
| 布隆过滤器                      | 一种内存高效的索引数据结构                            |
| 疾病控制中心                     | 内容定义的分块:将数据分成更小的块基于数据特征的组块[14]           |
| 英国国教会                      | 用于安全重复数据消除的聚合加密[56]                      |
| 矮胖的人或物                     | 重复数据消除的最小存储单元                            |
| 容器                         | 固定大小的存储单元,用于存储经过重复数据消除的数据块,通常大小为几兆字节[13] |
| DDFS                       | Data Domain 文件系统[13]                     |
| 指纹                         | 块或文件的加密安全散列摘要,例如SHA1                     |
| 乔治勋章                       | 垃圾收集:删除后回收可用空间操作                         |
| 硬盘驱动器                      | 硬盘驱动器                                    |
| LBFS                       | 低带宽网络文件系统[14]                            |
| landingzone 登陆区            | 伦佩尔-齐夫:通用压缩算法[1617]                      |
| 拉宾                         | 用于疾控中心的拉宾哈希算法[77]                        |
| (同 solid-statedisk) 固态(磁)盘 | 固态硬盘                                     |
| 超大杯                        | 基于重复数据消除的归档存储系统[11]                      |



夏雯于2014年获得中国华中科技大学(HUST)计算机科学博士学位。他目前是HUST大学计算机科学与技术学院的助理教授。他的研究兴趣包括重复数据删除、数据压缩、存储系统、云存储等。他在主要期刊和国际会议上发表了20多篇论文,包括IEEE-TC、IEEE-TPDS、USENIX ATC、USENIX FAST、INFOCOM、C、MSST、IPDPS、热存储等。他是ACM、CCF和IEEE



洪江于1982年获得中国武汉华中科技大学计算机工程学士学位；1987年获得加拿大多伦多大学计算机工程硕士学位；1991年获得美国得克萨斯大学A&M学院计算机科学博士学位。他现在是德克萨斯大学计算机科学与工程系的教授。

在此之前，他曾担任美国国家科学基金会的项目主管。自1991年以来，他一直在内布拉斯加-林肯大学担任威拉·凯瑟教授。他已经毕业了13名博士生，他给予博士学位的美国机构获得学术终身职位，要么受雇于美国主要的信息技术公司。他目前的研究兴趣包括计算机体系结构、计算机存储系统和并行输入输出、高性能计算、大数据计算、云计算、性能评估。他最近担任IEEE并行和分布式系统事务的助理编辑。他在这些领域的主要期刊和国际会议上发表了200多篇文章，包括IEEE-TPDS、IEEE-TC、IEEE会议录、ACM-TACO、JPD、ISCA、MICRO、USENIX ATC、FAST、EUROSYS、LISA、SIGMETRICS、ICDCS、IPDPS、MIDDLEWARE、OOPAS、ECOOP、SC、ICS、HPDC、INFOCOM、ICPP等。他的研究得到了国家自然科学基金会、国防部、得克萨斯州和内布拉斯加州的支持。蒋博士是美国电气与电子工程师协会会员，美国计算机学会会员。



冯丹分别于1991年、1994年和1997年获得中国华中科技大学(HUST)的计算机科学与技术学士、硕士和博士学位。她是HUST计算机科学与技术学院的教授和院长。她的研究兴趣包括计算机体系结构、海量存储系统和并行文件系统。她在主要期刊和国际会议上发表了100多篇文章，包括IEEE-TC、IEEE-100

快速运输服务、JCS、快速、USENIX空管、国际呼叫中心、国际呼叫中心。她担任多个国际会议的项目委员会成员，包括2011年、2013年和2012年、2015年的MSST会议。她是IEEE成员，也是ACM成员。



弗雷德·道格拉斯分别于1987年和1990年获得加州大学伯克利分校的计算机科学硕士和博士学位，并于1984年获得耶鲁大学的计算机科学学士学位。自2009年以来，他一直在EMC工作，专注于系统和存储技术，如闪存、重复数据消除、压缩、负载平衡等。他曾在其他工业应用研究机构工作，包括松下电器、美国电话电报公司(贝尔)实验室和IBM Research，并且是客座教授。

普林斯顿大学。他在主要期刊和会议上撰写了50多篇论文，如美国计算机学会ToS、FAST、USENIX空管、美国计算机学会中间件、万维网等，他是大约60项已发布美国专利的发明人。他在2007-2010年间担任IEEE互联网计算主编，并自1999年以来一直担任其编辑委员会成员。他是2016-2018年IEEE计算机学会理事会成员，IEEE计算机事务副主编，IEEE高级成员，ACM和USENIX成员。



菲利普·希尔分别于2000年和2001年获得斯坦福大学计算机科学学士和硕士学位。他还分别于2004年和2008年获得普林斯顿大学计算机科学硕士和博士学位。自2007年以来，他先后在Data Domain和EMC工作，在一家CTO组织中从事计算机存储系统、重复数据消除、压缩、数据表征、闪存缓存和非易失性存储器领域的研究和高级开发。他有更多

在期刊和会议上发表了25篇以上的文章，包括ACM ToS, ACM ToG, SIGGRAPH, USENIX ATC, FAST, 中间件, SMI, MSST, LISA, HotStorage等。他是30多项专利的发明者。



余华分别于2001年和2005年获得中国武汉大学计算机学学士学位和博士学位。他是中国华中科技大学的教授。他的研究兴趣包括计算机架构、云计算和网络存储。他在主要期刊和国际会议上发表了60多篇论文，包括IEEE计算机事务(TC)，IEEE并行和分布式系统事务(TPDS)，USENIX

M、SC、ICDCS和MSST。他曾在多个国际会议的项目委员会成员，包括INFOCOM、RTSS、ICDCS、MSST、ICNP、ICPP、IWQoS。他是IEEE和CCF成员。



傅敏目前是华中科技大学计算机体系结构专业的博士生。他目前的研究兴趣包括数据重复数据消除、存储系统和可靠性。他在主要期刊和会议上发表了几篇论文，包括IEEE-TPDS、USENIX空管、FAST等。



张玉成目前是中国华中科技大学(HUST)计算机架构专业的博士生。他的研究兴趣包括数据复制、存储系统等。他在被引用的期刊和会议上发表了几篇论文，包括IEEE-TC、INFOCOM等。



周玉坤目前是中国华中科技大学(HUST)计算机体系结构专业的博士生。他的研究兴趣包括数据复制、存储安全等。他在包括PEVA、MSST等在内的知名期刊和会议上发表多篇论文。