纠删码存储系统中数据修复方法综述

摘　要　作为当今存储系统中常见的容错选择，纠删码技术具有存储开销低的优势，然而在进行数据修复时仍然一些流量和性能方面的问题。近年来设计了各种纠删码和修复算法来缓解修复流量和提高性能，并通过给出纠删码技术中数据修复完成时间的计算模型，指出影响修复性能的关键因素，进而选取计算开销、读写开销、传输开销作为修复性能的评价标准，重点讨论了其关键性技术的优缺点;最后从修复性能、可靠性、存储开销等方面对现有编码方案进行对比，并指出未来可能的研究方向。

关键词　纠删码;修复算法;数据修复;性能优化;存储系统;多副本

Summary of data repair methods in erasure correcting code storage system

Abstract　As a common fault-tolerant choice in today's storage systems, erasure correction code technology has the advantage of low storage overhead, but it is still hindered by the introduction of non constant traffic when repairing data. In recent years, various erasure codes and repair algorithms have been designed to alleviate the repair traffic. By giving the calculation model of the completion time of data repair in erasure code technology, the key factors affecting the repair performance have been pointed out, and then the calculation cost, read and write cost, and transmission cost have been selected as the evaluation criteria of the repair performance. The advantages and disadvantages of its key technologies have been emphatically discussed; Finally, the existing coding schemes are compared in terms of repair performance, reliability, storage overhead, etc., and the possible research directions in the future are pointed out

Key words　Erasure correcting code; Repair algorithm; Data repair; Performance optimization; storage system

1 引言

现存的存储系统通常由大量存储节点（简称为“节点”）组成，以适应爆炸式增长的数据量，从而使故障意外出现，且这种问题普遍存在。为了在出现故障时保护数据可靠性，许多存储系统采用复制和数据编码，两者都依赖于预存储附加冗余以修复丢失的数据。与简单地存储相同副本的复制不同，纠删码确实可以以更少的存储消耗达到相同的容错程度，因此在商品存储系统中更为可取。原则上，纠删码包括两个轻量级计算操作，即命名编码（基于原始数据块生成冗余块）和解码（基于幸存的块来修复原始数据块），从而实现原始数据与冗余之间的有效转换。

尽管纠删码具有存储效率，但它往往需要检索并复用幸存的块以修复单个块，因此很可能会产生大量的修复流量（即，通过网络传输以进行修复的数据量）。为了解决修复中的 I/O 放大问题，现有研究主要采用以下方法：（I）构造具有可证明减少的修复流量的新理论纠删码，有局部可修复码、旋转 RS 码和再生码；（ii）设计有效的修复算法以并行化修复过程；以及（iii）利用基于机器学习的预测技术，在故障发生之前，利用修复算法主动重新存储数据。

多副本技术通过将数据的多个冗余副本分布在不同机器上，当存储节点发生故障时，自动将服务切换到其他副本，从而实现数据的高可靠和高可用。然而随着数据的持续增长，在 PB 级别的数 据中心，多副本技术会引入极大的存储开销。面对如此情况，纠删码技术因其低存储开销的 特点近年来受到越来越多的关注。其中，RS 编码(Reed-Solomon code)是目前被广泛使用的纠删码方案，它将 k 个数据块按照一定的编码规则，生成 m 个校验块，对于这 k+m 个编码块，其编码性质保证通过任意的 k 个编码块均能重建出所有数据。以 RS(4,2)编码为例，它只需占用 1.5 倍的存储空间，就具有与三副本技术相同的容错能力。然而当数据丢失时，纠删码技术数据修复的速度没有多副本快。为了修复丢失的数据，多副本技术只需拷贝对应数据的冗余副本。而纠删码需要读取 k 块数据，通过计算重建丢失的数据，相比之下，占用了更多

的磁盘带宽和网络带宽，也引入了额外的计算开销。这不仅导致修复时间过长，而且也对前台应用带来干扰。

针对纠删码修复存在的性能缺陷，近年来涌现出很多理论设计和工程实现的工作。本文将聚焦于纠删码数据修复，从系统性能优化的一般性原则出发，分析影响纠删码数据修复性能 的关键因素，如图 1 所示，围绕读写、计算、传输 3 个层级，对现有的优化技术进行讨论。其中，从减少计算量和加速计算执行两个角度介绍了计算优化方面的工作;从降低 I/O 总量和提高 I/O 并行度两个角度介绍了读写优化方面的工作;从减少单个节点的传输量和减少参与修复节点数量两个角度介绍了传输优化方面的工作。最后，对比分析了针对修复性能优化的编码方案，并指出未来可能的研究方向。

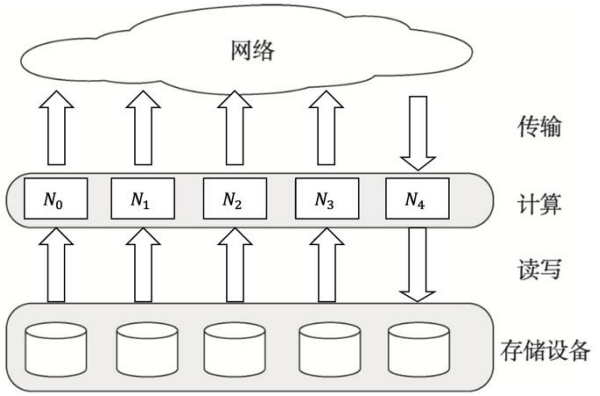

图 1　修复性能优化层级图

2 计算优化

RS 编码的运算法则建立在迦罗瓦域 GF(2^w)上，修复过程不仅涉及残余矩阵的求逆运算，同时涉及 复杂的有限域运算。GF(2^w)上的加法等同于异或运算，而乘法的计算相对复杂，涉及多项式乘法和取余运算。目前针对计算的优化主要分为两类:减少计算量和加速计算执行。

2.1　减少计算量

在柯西编码的生成矩阵中，“1”的数量反映计算过程中需要的异或次数，因此大量的研究工作试图通过寻找低密度生成矩阵的编码方案来减少计算量。然而对于低密度的编码矩阵，其解码矩阵可能非常稠密，无法在修复时也同样减少计算量。

为了降低矩阵密度对计算量的直接影响，已有工作采用计算调度的方式来减少计算过程中的计算量。其基本思想是调整计算的执行顺序，最大可能地利用计算过程中的中间结果来减少重复的计算。如图 2 所示，在 B=AX 的计算中，为了计算 B 中 4 个目标块的值，传统的计算方式如图 2 中右边所示，整个计算需要 12 次异或操作。如果利用 b2 =b1+x1，可以将整体的操作次数从 12 降低至 9。然而，如何根据给定的生成矩阵寻找最优的计算调度方案，已经被 Huang 等人推测是 NP 完全问题，现有的解决方式主要基于贪心原则和启发式搜索寻找次优的调度方案。

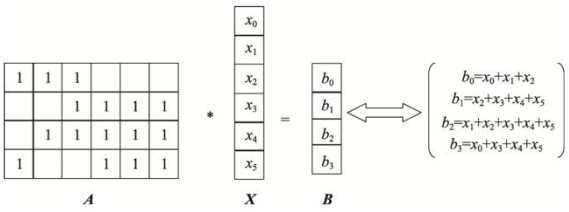

图 2　01 矩阵下的数据编码过程

2.2　加速计算执行

有限域的乘法运算存在计算复杂的问题，为了加快执行速度，目前有两种优化方式:位运算转换法 和查表法。位运算转换法基于有限域上的元素都 可以采用二进制矩阵进行表示，它将生成矩阵转换为 GF(2)域空间上的 01 矩阵，因此整个计算过程只存在异或运算，降低了计算的复杂度。

查表法是工程实现中一种常见方式，它通过预先生成有限域乘法表，在计算过程中避免复杂的多项式计算，只需查询相应的计算结果。在 GF(2^w)中，二维乘法表需要耗费 O((2^w)^2)的存储空间，随着 w 的增大，该方式不再适用。由于有限域的非零元素均能用本原元的指数形式表示，进而有限域的乘法能够转换为指数上的加法运算。因此，在选定本原元的情况下，借助对数表(数字到指数的映射)和反对数表(指数到数字的映射)，两数的乘法运算只需两次对数表查询得到相应的指数数值，然后通过求和运算获得乘法结果的对应指数数值，最后通过一次反对数表查询，即可得到乘法运算的结果。该方式只需 O(2^w)的存储空间。查表法用访存操作替代计算，将执行速度的瓶颈从处理器转移到内存。

由于 RS 编码中涉及大量的矩阵向量乘运算，相比 CPU，GPU 的数据矢量化和并行处理能力更加适合这种计算模型。

3 读写优化

磁盘读取的时间开销受读取的数据总量和并行度控制，通过降低 I/O 总量和提高 I/O 并行度两种方式，均可以有效降低单盘负载，从而减少磁盘读取的时间开销。

3.1 降低I/O总量

阵列码 EVENODD 和 RDP 是常见的 RAID6(re-dundant arrays of independent disks)编码，在其编码布局中均存在水平校验组和对角线校验组。当单个数据盘出错时，传统修复方式采用水平校验组对数据进行修复。如图 3(a)所示的 RDP(6,2)数据布局，一共 6 块磁盘，其中数据盘 4 块。当磁盘D_0损坏时，采 用水平校验组，需要从D_1、D_2、D_3、D_4读取 16 块数据，该修复方式不仅未使用磁盘D_5，同时忽略了两种校验组间存在的数据重叠。另外存在一种对 RDP 恢复进行了优化且是首次提出同时使用两种校验组的混合修复的方式。如图 3(b)所示，对于丢失的 4 个数据块，其中一半数据块的修复采用水平校验组，另一半数据块修复采用对角线校验组。该修复方式下，读取的数据块存在 4 块重叠，需要读取的数据块数量降低为 12，从而减少了 25%的 I/O 读取总量。混合修复方式通过寻找读取过程中最大的重叠区域来降低数据读取总量，基于同样的思想，存在分别针对 EVENODD 编码、X-Code 编码和 HDP(horizontal-diagonal parity)编码提出了单节点故障下的快速恢复算法。然而，上述的双容错编码由于其特定的数据布局，只需考虑两种校验组，当将该方式扩展到容错能力为 3 的 STAR 编码或者更通用的柯西 RS 编码时，复杂的数据布局导致使用指数级的搜索时间来寻找最大的重叠区域。

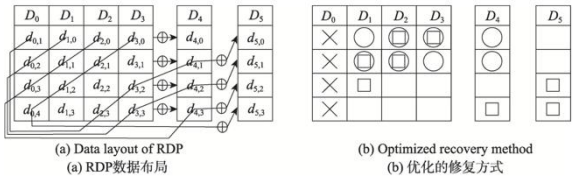

图 3　RDP(6,2)编码的数据布局及修复方式

3.2 提高I/O并行度

传统磁盘阵列在运行过程中，为了及时替换出错的磁盘，往往配备了热备盘。然而在没有发生磁盘失效的正常状态下，热备盘只能空闲等待。针对这个问题，有人提出了分布式空闲块的思想，不再设置专门的热备盘，而使用磁盘中的空闲区域作为修复使用的存储空间。在正常状态下，该方式让热备盘也加入使用，所有的磁盘都能参与服务，提高了 I/O 处理能力;失效修复时，更多磁盘的参与降低了修复时间。

除了热备盘的利用，分簇技术(parity declus-tering)在恢复数据读取总量不变的情况下，通过使用更多的磁盘，显著降低了修复的时间开销。以 S2-RAID 编码设计为例，S2-RAID 在条带大小不变的情况下，允许加入更多的磁盘参与修复。

4 传输优化

传统 RS(k,m)编码对 k 个数据块生成 m 个校验块，在修复某块数据时需要将 k 个参与节点上的数据传输到修复节点进行数据重建。以 RS(8,2)为例，数据块大小为 128MB，在数据修复过程中，整个网络需要传输 1GB 的数据量。这给分布式存储系统带来了较大的网络负担。目前，传输优化主要有减少单个参与节点传输量和减少参与节点数量两种方式。

4.1 减少单个参与节点传输量

RS 编码将节点上存储的数据作为单一整体进行计算，因而修复过程需要参与节点读取并传输所有数据。如果将节点上的数据划分成子块，以子块作为最小编码单位，则参与节点可以只传输部分子块 数据用于修复。

为了降低单个参与节点的数据传输量，有人提出了一类新型编码——再生码(regenerating code，RGC)，是参数为(n,k,d)的再生码，满足任意 k 个节点能够重建整个数据，单节点出错时，从任 意 d(k≤d≤n-1)个节点上传输数据进行修复。与 RS 编码相比，虽然 RGC 编码需要更多的节点参与修复，但是每个节点的数据传输量低，从而能够降低整体的网络传输量。目前，RGC 编码主要关注两类：MSR(minimum storage regenerating)编码和 MBR(minimum bandwidth regenerating)编码。

MSR 编码与 RS 编码同等存储开销下，具有更低的网络开销，而 MBR 编码允许存储更多的数据来获得更低的网络开销。

4.2 减少参与节点数量

传统 RS 编码将多个数据块通过线性组合的方式生成全局校验块，数据修复时需要读取的数据量与 参与构建的数据块数量相等。在全局校验块数量固定的情况下，增加全局校验组中数据块的数量，能够 降低编码的存储开销比率，却会导致参与修复的节点数量增多，带来网络传输开销和计算开销的增加。局部修复码(local reconstru-ction code，LRC)在传统 RS 编码的基础上，引入局部分组的思想，将编码 块划分成多个小分组，分组内部生成局部校验块进行容错保护。数据修复不再完全依赖于全局校验组，优先使用局部分组内的数据进行重建，从而通过减少参与节点的数量，降低了修复过程中的网络传输开销和磁盘读取开销。

5 讨论

服务领域的存储系统需要提供连续不停机的存储服务，因此数据修复操作必须在线进行。一方面，基于纠删码的存储系统的修复完成时间受到计算开销、磁盘读写开销和网络传输开销等因素的共同影响。另一方面，因为数据修复操作和前台 I/O 操作共享甚至竞争存储系统的 I/O 资源，数据修复操作对计算资源、磁盘带宽和网络带宽的占用又会给前台应用性能带来明显干扰。因此，计算开销、读写开销和传输开销是影响纠删码修复性能的三大关键因素。

为了提高纠删码存储系统中数据修复效率，研究者主要从计算优化、读写优化和传输优化等三方面开展工作。

(1)计算优化方面:计算硬件的发展允许通过 SIMD 技术实现数据级并行，允许对多个数据单元同时执行相同的操作，显著加快了基于有限域运算的编码计算。而通过调整计算的执行顺序，避免了计算过程中不必要的重复计算，能够有效减少计算量，进一步降低计算开销。

(2)读写优化方面:通过合理选择修复使用的条带，让数据恢复所需的数据出现重叠，使得读取同一块数据能够用于多块数据的修复，减少了数据读取总量。另一方面，在不减少数据读取总量的情况下，通过将更多磁盘引入到数据恢复中来，降低单盘上的读取负载，从而减少整体的读取时间。

(3)传输优化方面:为了降低网络传输量，研究者分别设计出 RGC 编码和 LRC 编码。RGC 编码通 过降低单个参与节点的数据传输量，减少修复过程中的网络开销。LRC 编码通过引入局部分组，解除了修复参与节点数量与全局校验组之间的关联，以存储开销的增加换取修复参与节点数量的减少，从而降低修复过程中的磁盘读写开销和网络传输开销。

6 总结和展望

本文分析了影响纠删码修复的关键因素，从计算、读写、传输三方面对优化纠删码修复性能的关键技术进行了探讨。现有的编码方案中大多未能兼顾计算、读写、传输三方面，如何从理论设计和系统实现上优化实现这三目标，还存在巨大的技术挑战。

理论设计方面，同等容错能力下，MSR 编码能够 在不增加额外存储开销的情况下，显著降低网络传输量。但是 MSR 编码在高容错能力时还存在参数选 取受限，计算复杂和存储利用率过低等缺陷。如何设计高编码率、高可靠和低复杂度的 MSR 编码将会是未来的一个重要研究方向。

系统实现方面，结合设备特性，对编码方案进行适配和调优。编码的理论效果与工程实现存在巨大的鸿沟，以磁盘读取为例，再生码修复过程需要完整读取磁盘上的数据，是否存在只读取部分子块的可能性?如果存在,当将原有的大块连续读取转变为多次跳读之后，虽然读取总量有所降低，但其带宽却可能下降。针对不同的编码方案，如何在工程实现 中对计算、读取、传输三方面进行适配和调优，也将是未来研究的重点。