◆大数据技术◆

纠删码存储系统中数据修复性能优化研究进展与展望*

沈佳杰,向　望**,沈敏虎,武博淳,赵泽宇,张　凯
(复旦大学,校园信息化办公室,上海　200433)

摘要:纠删码被广泛应用于分布式存储系统以保存在线应用的用户数据。当部分存储节点发生故障时,纠删码存储系统需使用新的存储节点替换原有失效节点,并恢复失效的用户数据。由于需要执行数据编码、传输和读写等操作,纠删码存储系统通常需要消耗较长的时间执行数据修复操作,存储的用户数据将长期处于不可靠状态。为了保障存储数据的可靠性,研究学者提出了多种数据修复性能优化方案以减少数据修复时间。本文介绍了数据修复性能优化问题,分析了各个应用场景下主要的性能瓶颈和性能优化难点,总结了提升数据修复性能的主要技术方案和研究工作,并对数据修复性能优化研究领域的未来发展方向进行展望,为纠删码存储系统设计人员准确选择适合特定应用场景的数据修复性能优化方案提供思路。

关键词:分布式存储系统;纠删码存储系统;数据修复操作;数据可靠性;性能优化

中图分类号:TP391　　**文献标识码**:A　　**文章编号**:1005-9164(2023)01-0149-20
DOI:10.13656/j.cnki.gxkx.20230308.017

纠删码被广泛部署于分布式存储系统以保存在线应用的用户数据[1]。为了保障数据的可靠性,纠删码存储系统[2]将用户数据切分成多个数据分片,编码生成校验分片,存储到多个存储节点[3]。当部分存储节点发生故障时,纠删码存储系统需使用新的节点替换失效节点,以修复失效的数据。

随着分布式存储系统规模的增大,存储节点频繁失效已成为常见的问题。在单个 Facebook 数据中心 3 000 个存储节点的集群中,每日至多有 110 个失效节点引发数据修复事件[3]。为了保障存储数据的可靠性,纠删码存储系统需要快速修复失效节点存储的数据[4]。然而,随着存储节点保存数据量的增加,修复存储节点失效数据的时间也随之增加[5]。

在数据修复过程中,纠删码存储系统需要执行数据读写、编码和传输操作,通常需要消耗较长的时间来修复失效数据[6]。在存储节点长时间无法得到修复的情况下,分布式存储系统处于不稳定的状态,严重影响存储数据的可靠性[7]。研究学者针对不同纠删码存储应用场景提出了大量高效的数据修复方案。纠删码存储系统设计人员需要充分了解各种数据修

收稿日期:2022-10-10　　修回日期:2022-10-19
*国家自然科学基金项目(61971145)和中国高等教育学会重大项目(2020XXHZ01)资助。
【第一作者简介】
沈佳杰(1989-),男,工程师,主要从事纠删码存储系统和下一代校园网络结构研究。
【**通信作者】
向　望(1981-),男,高级工程师,主要从事无线网、物联网和下一代互联网研究,E-mail:xiangw@fudan.edu.cn。
【引用本文】
沈佳杰,向望,沈敏虎,等.纠删码存储系统中数据修复性能优化研究进展与展望[J].广西科学,2023,30(1):149-168.
SHEN J J,XIANG W,SHEN M H,et al. Research Progress and Prospect on Performance Optimization of Data Recovery for Erasure-Coded Storage Systems [J]. Guangxi Sciences,2023,30(1):149-168.

复方案的特性,根据应用场景的特性部署相应的数据修复方案[8]。

为帮助设计人员快速了解数据修复方案,本文介绍了纠删码存储系统的数据修复性能优化问题,分析了磁盘阵列存储系统、分布式存储系统和云际存储系统 3 种应用场景下数据修复操作存在的主要性能瓶颈、亟需优化的性能指标以及优化性能指标存在的技术难点,综述了纠删码存储系统主要的应用场景以及在这些场景下制约数据修复操作性能的主要因素,并对当前采用的性能优化技术方案和研究工作进行总结,从学术研究和工业应用领域展望了数据修复技术未来的发展方向。

1 数据修复性能优化现状

1.1 纠删码存储系统主要应用场景

为了保证在部分节点失效的情况下存储数据的可靠性,纠删码存储系统将用户数据切分成多个数据块,将数据块编码生成多个校验块,然后将这些数据块和校验块保存到多个存储节点[3]。

纠删码存储系统的数据写入过程如图 1 所示。纠删码存储系统将用户数据分片生成数据块 A 和数据块 B,编码生成校验块 $A+B$ 和校验块 $A+2B$,并将这些分片存储到存储节点。当任意两个节点失效时,纠删码存储系统可以通过剩余节点数据恢复失效节点数据。

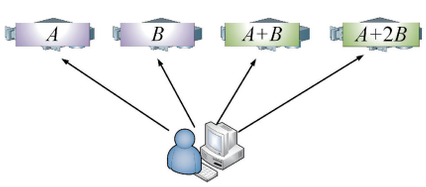


图 1 纠删码存储系统数据写入过程[9]

Fig. 1 Process of write operations in erasure-coded storage systems[9]

在消耗较少的存储开销的前提下,纠删码存储系统能有效保障存储数据的可靠性,因此被广泛部署到多种应用场景,用于存储用户数据[9]。纠删码存储系统主要部署到以下应用场景。

①磁盘阵列存储系统[10]。由于存在磁盘失效的情况,单台服务器需要使用多块磁盘组成磁盘阵列存储系统,以保存用户数据[11]。当磁盘失效时,磁盘阵列存储系统需要替换失效磁盘,根据编码规则恢复出失效磁盘的数据并存储到新的磁盘中[12]。

②分布式存储系统[3]。为了应对存储节点失效的问题,分布式存储系统将用户数据编码后存储到多个存储节点。当存储节点失效时,分布式存储系统将替换失效节点,读取幸存节点中的数据,恢复失效的用户数据至替换的新生节点上[3]。

③云际存储系统[13]。由于单个数据中心存在用户数据失效的问题,云际存储系统将用户数据加密编码后存储到多个云存储节点。云际存储系统能保障在部分存储节点失效的情况下恢复失效数据。与此同时,云际存储系统可以保证在黑客非法访问部分节点数据的前提下,无法获取用户保存的原始数据[14]。

1.2 纠删码存储系统数据修复过程

当存储节点失效时,需要使用新的存储节点替换失效节点,以修复失效节点存储的数据。纠删码存储系统的数据修复过程如图 2 所示。纠删码存储系统需要重新下载数据块 A 和数据块 B,编码生成校验块 $A+B$。然而,随着存储系统规模的扩大和单个存储节点保存数据量的增加,数据修复操作通常需要消耗较长的时间来恢复失效的数据,这将造成纠删码存储系统长时间处于不可靠的状态,减少数据修复操作的时间成为保障分布式存储可靠性的关键性因素[3]。

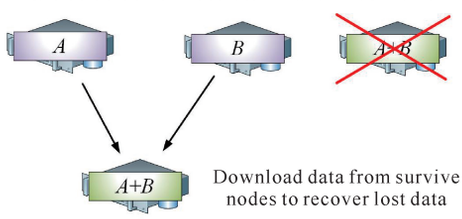


图 2 纠删码存储系统的数据修复过程[3]

Fig. 2 Data recovery process in erasure-coded storage systems[3]

1.3 各种应用场景下的性能瓶颈

在数据修复过程中,纠删码存储系统需要执行数据读写、编码和传输操作。在各种应用场景下,限制数据修复操作性能的因素通常有较大的差异性[15]。图 3 展示了磁盘阵列存储系统、分布式存储系统和云际存储系统的层次关系。

在数据修复过程中,纠删码存储系统主要对以下性能开销。

①编码计算开销。在数据修复过程中,纠删码存储系统需要编码用户数据以恢复失效的数据,会引起较大的编码计算开销。

②磁盘读写开销。在数据修复过程中,需要读写存储在磁盘中的编码数据。由于磁盘读写速度通常远小于编码计算速度,大量的磁盘读写操作会增加数据修复所消耗的时间。例如,在磁盘阵列存储系统中,应尽可能减少数据修复过程中磁盘读写的数

据量。

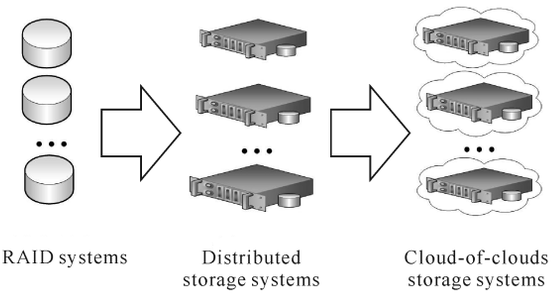


图3 磁盘阵列存储系统、分布式存储系统和云际存储系统的层次关系[15]

Fig. 3 Hierarchical relationship between RAID systems, distributed storage systems, and cloud-of-clouds storage systems[15]

③网络传输开销。由于存储节点之间需要传输数据,当执行大数据修复操作时,通常会增加数据传输开销。例如,在云际存储系统中,各个云存储节点之间通过互联网传输数据,通常会成为数据修复操作的性能瓶颈。

表1展示了磁盘阵列存储系统、分布式存储系统和云际存储系统执行数据修复操作的修复对象和性能瓶颈。存储系统设计人员需要根据应用场景部署相应的数据修复方案。在实际纠删码存储系统中,存储节点状态、数据放置位置、数据编码方案等都会影响执行数据修复操作的性能。研究学者需要综合数据修复过程中的难点问题,在保证存储数据可靠性的前提下减少纠删码存储系统执行数据修复操作带来的开销。

表1 各种应用场景执行数据修复操作的修复对象和性能瓶颈[15]

Table 1 Repair objects and performance bottleneck in different application scenarios[15]

应用场景 Application scenarios	修复对象 Repair objects	性能瓶颈 Performance bottleneck
RAID systems	Disk and data volume	Disk I/O performance
Distributed storage systems	Storage node	Network bandwidth between storage nodes
Cloud-of-clouds storage systems	Cloud storage node	Network bandwidth between data centers

1.4 数据修复性能优化过程存在的难点

为了快速完成数据修复过程,纠删码存储系统需要使用高效的数据编码和传输方案[16]。纠删码存储系统数据修复方案性能优化面临以下难点问题。

①存储系统状态管控[17]。随着存储节点数量的增加,大规模纠删码存储系统管控存储节点的难度随之增加[18]。为了保障存储用户数据的可靠性,纠删码存储系统需要准确获取各个存储节点的状态,快速定位纠删码存储系统的失效存储节点。

②数据放置方案优化[19]。由于存储节点在网络资源、存储能力和计算资源等方面具有较大的差异性,纠删码存储系统需要综合考虑存储节点的特性,选择合适的存储节点放置数据,提高存储节点失效时数据修复操作的执行效率[20]。

③数据编码方案优化[21]。纠删码存储系统需要将用户数据编码后存储到多个存储节点,数据编码方案会影响纠删码存储系统执行数据修复操作的效率[22],高效的数据编码方案更能保障数据修复操作的性能[23]。

④数据修复过程优化[4]。由于存储节点之间的网络带宽等资源通常十分有限且具有较大的异构性[6],纠删码存储系统需要读取幸存节点存储的数据,并通过合理方案高效地将数据传输到新生节点[16],从而恢复失效数据。为了提升数据修复操作的性能,数据修复需要高效的数据读写和传输方案来提高数据修复操作的效率[24]。

表2展示了各个应用场景下解决数据修复性能优化关键性问题的技术难点。

1.5 数据修复性能优化关键技术

为了解决数据修复过程存在的难点问题,研究学者提出多种技术方案来提升纠删码存储系统数据修复操作的性能。

①存储系统状态管控技术。由于分布式存储系统通常包含大量的存储节点,纠删码存储系统通过软件定义存储采集节点的运行数据,从而准确获取各个存储节点的运行状态。

②存储系统数据放置技术。由于存储节点的网络带宽资源存在差异性,纠删码存储系统需要将编码数据合理地放置在各个存储节点,保障纠删码存储系统能高效地执行数据修复操作[25]。此外,由于在数据修复过程中存储节点之间通常需要传输数据,纠删码存储系统需要根据当前节点之间的网络状态迁移存储节点中保存的数据,以平衡存储节点的数据读写负载[26]。

③高效数据编码方案技术。为了保证存储数据的可靠性,纠删码存储系统需要编码用户数据,将编码后数据保存到多个存储节点。通过构建高效的数据编码方案,纠删码存储系统能有效减少数据修复过

表 2　数据修复性能优化关键性问题的技术难点[15]

Table 2　Technical difficulties in key issues of performance optimization for data recovery process[15]

应用场景 Application scenarios	磁盘阵列存储系统 RAID systems	分布式存储系统 Distributed storage systems	云际存储系统 Cloud-of-clouds storage systems
State management of storage systems	Obtain the disk state, predict the disk failure probability	Obtain the state of the storage nodes and network	Obtain the network state between cloud storage nodes
Optimization of data placement schemes	Improve the parallel I/O performance and scalability	Fully utilize network bandwidth between storage nodes	Fully utilize the network bandwidth between clouds
Optimization of date coding schemes	Reduce the computation and disk I/O overhead	Reduce the disk I/O and transmission overhead	Reduce the network traffic across data centers
Optimization of data recovery process	Schedule coding sequence to reduce disk I/O overhead	Select the proper transmission scheme for data recovery process	Optimize the transmission process between storage nodes

程中磁盘读写数据量和网络带宽资源的消耗量,提升数据修复操作的性能。

④数据修复过程性能优化技术。在数据修复过程中,存储节点之间需要从幸存节点中读取数据,编码恢复失效数据,将这些恢复的数据存储到新的节点。根据存储节点和网络状态,纠删码存储系统需要调整数据修复操作的执行策略,保障存储数据的可靠性。

纠删码存储系统性能优化关键技术的作用如图 4 所示。

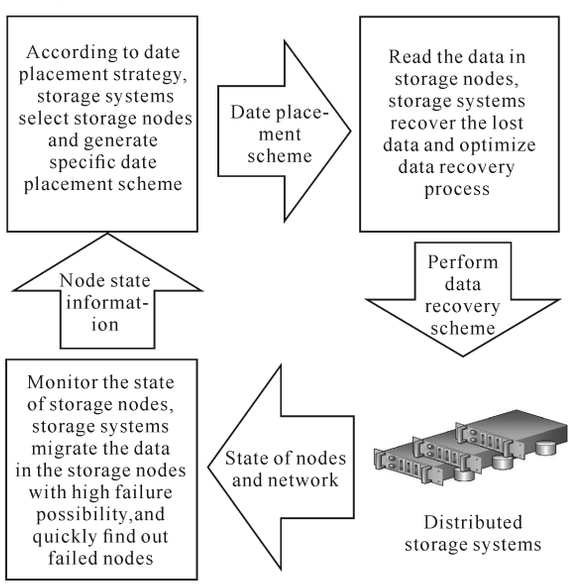


图 4　纠删码存储系统性能优化关键技术的作用

Fig. 4　Role of key technologies in performance optimization of erasure-coded storage systems

2　数据修复关键技术实现方案

2.1　存储系统状态管控技术

根据存储设备的运行数据,纠删码存储系统可以预测存储设备未来的状态。图 5 使用常见的磁盘阵列存储系统中磁盘故障预测来说明存储设备状态的

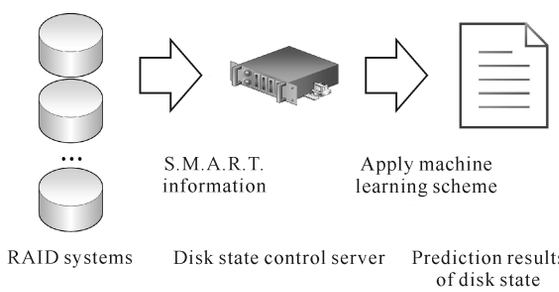


图 5　磁盘故障预测过程[17]

Fig. 5　Process of disk fault prediction[17]

预测过程。

纠删码存储系统主要通过以下步骤预测磁盘未来的故障情况。

首先,获取以往磁盘设备的数据。通过采集磁盘的状态数据,如 S. M. A. R. T.(Self-Monitoring, Analysis and Reporting Technology)日志数据[27],磁盘阵列存储系统能准确地获取磁盘通电时长、启动/停止技术、寻道错误率和读写错误率等信息。

其次,使用机器学习算法分析特定应用场景过往的磁盘状态数据,建立较为准确的磁盘失效故障模型,以预测未来可能发生的故障。常见的存储设备故障分为可预测故障和不可预测故障。虽然预测算法可以预测大量存储设备的失效情况,提前将数据迁移到新的存储设备中,从而减少数据修复操作过程带来的数据读写和传输开销问题,但是纠删码存储系统仍需提升数据修复操作的性能,以应对存储节点的不可预测失效情况。

最后,机器学习算法具有较强的应用场景针对性。由于当前机器学习普遍需要在一个特定的应用场景中采集较多的存储设备运行的日志数据,导致当前机器学习算法普遍难以适用于多种应用场景。为了提升各种应用场景下存储数据的可靠性,设计人员仍需引入其他的状态管控方案来保证纠删码存储系

统能准确地分析存储系统的运行状态。

2.2 存储系统数据放置技术

为了充分利用存储空间和存储节点之间的网络带宽资源，纠删码存储系统需要确保用户数据合理地放置在各个存储节点。在此基础上，纠删码存储系统通过并行化执行数据传输操作，将数据修复操作执行过程中的数据传输开销平均分配到各个存储节点，从而保障数据修复操作的性能[28]。纠删码存储系统数据放置过程如图 6 所示。

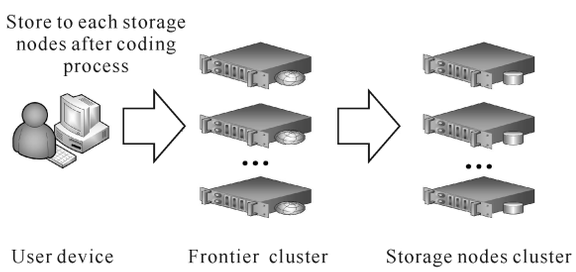


图 6 纠删码存储系统数据放置过程[18]

Fig. 6 Data placement process in erasure-coded storage systems[18]

此外，数据放置方案也需要满足纠删码存储系统数据读写操作的性能需求[29]。因此，存储设计人员需要综合考虑数据读写和修复操作的性能需求，从而部署适合的数据放置方案。本文使用一个 RAID+方案[20]下磁盘阵列存储系统数据放置的例子来说明编码后数据放置对数据修复操作执行效率的影响。

假设磁盘阵列存储系统包括 5 块磁盘，记为 D_0、D_1、D_2、D_3、D_4。使用条带为 3 的 RAID5 的编码方案，包括 2 个数据块和 1 个校验块。RAID+方案的数据修复过程如图 7 所示。图 7 使用红色、紫色和蓝色分别代表失效磁盘、尚未失效磁盘保存的数据和修复生成的数据。

根据正交拉丁方矩阵(Mutually Orthogonal Latin Squares，MOLS)[20]计算各个条带数据的放置位置，RAID+方案将编码后的用户数据均匀地分布在各个磁盘，以保障存储数据的可靠性。当磁盘出现故障时，RAID+方案首先将失效磁盘存储的数据并行化修复到尚未失效磁盘的预留空间。

在运维人员插入新的磁盘后，纠删码存储系统自动将修复完成的数据迁移到新添加的磁盘，通过并行化读写存储的编码数据以充分利用磁盘的读写带宽资源。RAID+方案能快速地恢复失效数据，保障大规模磁盘阵列存储系统执行数据修复操作的效率。

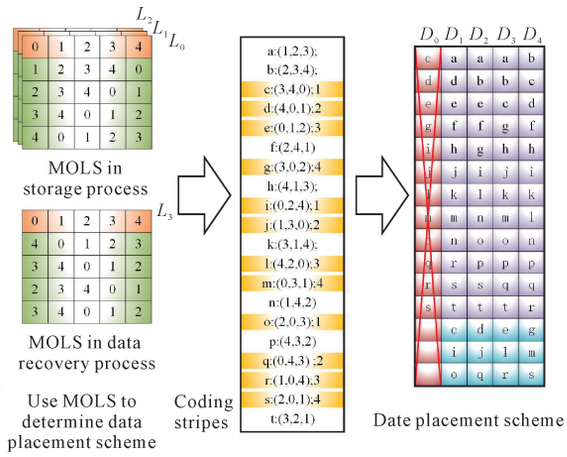


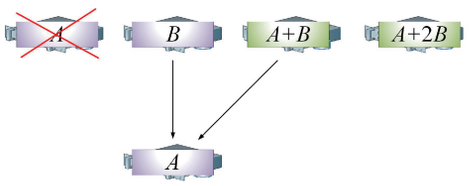
图 7 RAID+方案的数据修复过程示例[20]

Fig. 7 Example of data recovery process of RAID+ scheme[20]

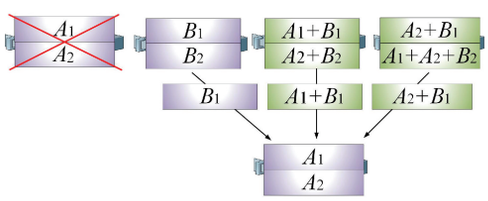
2.3 高效数据编码方案技术

由于用户数据编码后需保存到多个存储节点，因此纠删码存储系统需根据应用场景选择合适的数据编码方法来减少数据修复操作的开销。本小节以再生码(Regenerating Code，RC)数据修复过程为例，介绍数据编码方案对数据修复操作网络传输性能的影响[23]。

再生码对数据修复过程网络带宽消耗量的影响如图 8 所示。假设一个大小为 M 字节的文件被切分成数据块 A 和数据块 B，编码生成校验块 $A+B$ 和校验块 $A+2B$，则每个块的大小为 $\frac{M}{2}$ 字节。传统的 Reed-Solomon (RS)码[1]数据修复过程中，新生节点需要从任意两个存储节点各下载一个大小为 $\frac{M}{2}$ 字节的块，共计 M 字节数据[图 8(a)]。通过将数据块 A 和数据块 B 切分成数据分片 A_1、A_2、B_1 和 B_2，并编码生成对应的校验分片 A_1+B_1、A_2+B_2、A_2+B_1 和 $A_1+A_2+B_2$，每个分片的大小为 $\frac{M}{4}$ 字节。替换节点从每个幸存节点下载 1 个分片，共计下载 $\frac{3M}{4}$ 字节数据[图 8(b)]。



(a) Data recovery process of the RS code



(b) Data recovery process of RC

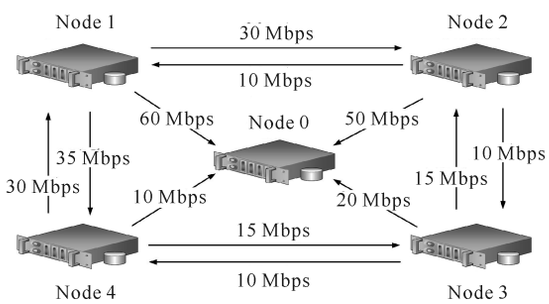
图 8　RS 码和 RC 对网络带宽消耗量的影响[25]

Fig. 8　Influence of RS code and RC on network bandwidth consumption[25]

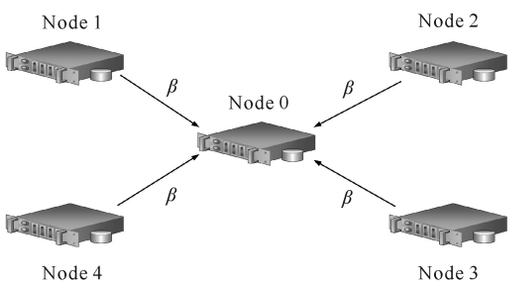
通过上述例子可知,纠删码存储系统使用再生码执行数据修复操作能有效减少传输的数据量。再生码可以保证数据修复过程中存储节点之间传输数据所消耗的网络带宽资源,能达到理论消耗量的下限[23]。在此基础上,高效的数据编码方案[21]也能有效减少数据修复过程中数据编码的计算开销和磁盘读写的数据量,提升数据修复操作的执行效率。

2.4　数据修复过程性能优化技术

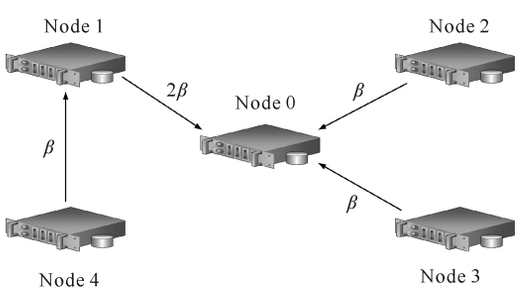
在数据修复过程中,纠删码存储系统需保障存储节点之间传输数据的效率。由于存储节点之间的网络带宽资源存在异构性,纠删码存储系统需要确定存储节点之间传输的数据量,以减少数据传输消耗的时间[22]。图 9 展示了存储节点之间传输的数据量对数据修复操作性能的影响。



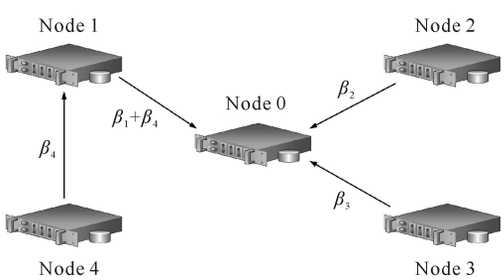
(a) Example of network topology



(b) Data recovery scheme with traditional star topology



(c) Data recovery scheme with tree topology



(d) Flexible data recovery scheme with tree topology

图 9　数据修复方案对数据修复操作性能的影响[6]

Fig. 9　Influence of data recovery scheme on data recovery operation performance[6]

图 9(a)展示了网络拓扑结构。假设需要修复一个 $M=480$ Mb 的文件,编码参数为 $d=4$ 个提供节点,$k=2$ 个数据节点,$n=5$ 个数据节点,将文件分成 $\frac{M}{k}=240$ Mb 分片。在执行数据修复操作过程中,新生节点需要从 4 个提供节点下载数据。

图 9(b)展示了传统星形数据修复方案。所有提供节点直接将数据发送给新生节点,新生节点需要从每个提供节点下载 $\beta=\frac{M}{k(d-k+1)}=80$ Mb 的数据,可得数据修复时间为 $\frac{\beta}{10\ \text{Mbps}}=8$ s。通过观察可以发现,存储节点 4 到存储节点 0 之间的网络带宽值最小,数据修复时间较长。

图 9(c)展示了基于树形拓扑的数据修复方案。基于树形拓扑的数据修复操作需要消耗 $\max\{\frac{2\beta}{60\ \text{Mbps}},\frac{\beta}{20\ \text{Mbps}}\}=4$ s。通过构建树形拓

扑,数据修复过程有效地避开了瓶颈链路,而相比于星形数据修复方案,有效减少了数据修复消耗的时间。

图 9(d)展示了基于树形拓扑的弹性数据修复方案。虽然树形拓扑在每一条链路上都传输相同的数据量,但是数据修复操作的执行效率仍受到传输路径上瓶颈网络带宽的影响。通过调整各个链路传输的数据量,基于树形拓扑的弹性数据修复方案能有效地减少数据修复消耗的时间。存储节点之间会传输 $\beta_1=90$ Mb、$\beta_2=90$ Mb、$\beta_3=60$ Mb 和 $\beta_4=90$ Mb 的数据量,执行数据修复操作所消耗的时间为 $\max\{\frac{\beta_1+\beta_4}{60\ \text{Mbps}},\frac{\beta_2}{50\ \text{Mbps}},\frac{\beta_3}{20\ \text{Mbps}},\frac{\beta_4}{30\ \text{Mbps}}\}=3\ s$。

由上述例子可知,通过调整存储节点之间传输的数据量,能充分利用云存储节点之间的网络带宽资源量,减少数据修复操作消耗的时间,提升数据修复操作的性能。因此,纠删码存储系统需要根据应用场景选择合适的数据修复方案来保障数据的可靠性。

2.5 数据修复关键技术实现方案分析

当存储节点失效时,纠删码存储系统需要选择合适的存储节点,编码生成待恢复的用户数据,以完成数据修复操作。在数据修复过程的各个阶段,部署数据修复性能优化方案能有效减少编码计算开销、磁盘读写开销和网络传输开销。

纠删码存储系统需要多种实现机制来减少数据修复过程的各种性能开销,数据修复关键技术优化各种性能开销的实现机制如表 3 所示。

表 3 数据修复关键技术优化各种性能开销的实现机制

Table 3 Key technologies of data recovery optimize the implementation mechanism of various performance overhead

关键技术 Key technology	编码计算开销 Coding calculation overhead	磁盘读写开销 Disk I/O overhead	网络传输开销 Network transmission overhead
State management of storage systems	Migrate the data with high failure probability to avoid unnecessary coding calculation overhead	According to the state of storage nodes, migrate the data to improve disk I/O performance during data recovery process	According to the network state migration data between nodes to improve network transmission performance
Optimization of data placement scheme	Select the proper storage nodes and parallelly perform coding operations	Reasonable place the data in storage nodes to speed up the I/O performance	Reasonable place the data in storage nodes to speed up network transmission process
Optimization of date coding scheme	Design coding scheme with low coding calculation overhead	Design coding schemes with low disk I/O overhead	Design coding schemes with low network transmission overhead
Optimization of data recovery process	Allocate the coding calculation overhead to multiple storage nodes	Schedule disk I/O process to reduce data volume	Schedule routing scheme to speed up data recovery process

3 数据修复关键技术相关研究进展

3.1 存储系统状态管控方案

研究学者引入大量的机器学习算法来预测即将发生的故障事件,以管控存储设备的运行状态。在分析大量站点磁盘 S. M. A. R. T. 日志数据的基础上,Lu 等[30]引入卷积神经网络-长短期记忆(Convolutional Neural Network Long Short-Term Memory,CNN-LSTM)预测数据中心的磁盘故障事件。Xu 等[31]利用磨损更新集成特征排序(Wearout-updating Ensemble Feature Ranking,WEFR)方案来选择 S. M. A. R. T. 特征,以提升预测闪存磁盘故障的正确性。Han 等[32]分析了不同在线应用场景负载下闪存磁盘失效情况之间的关联性。为了保障云计算服务的稳定性,Lin 等[33]提出 MING 框架结合多种机器学习算法,预测数据中心节点的失效事件。Li 等[34]利用 SystemX 来预测大规模云存储系统的节点故障。

为了保障纠删码存储系统的数据可靠性,研究学者设计了高效的纠删码存储系统管控机制。Lin 等[35]提出 RepairBoost 方案,使用有向无环图以充分利用节点之间的双向带宽资源完成数据修复操作。通过引入机器学习算法分析历史故障修复结果的数据,Li 等[36]设计了一种智能自动修复系统(Automated Intelligent Healing System,AIHS),通过选择合理的修复策略,实现大规模云服务系统故障的自我修复;Guo 等[37]在分析阿里数据中心时讨论了限制节点资源的原因;Cheng 等[38]使用纠删码来保障分布式机器学习算法运行的可靠性;Huang 等[39]提出 OmniMon 架构,使用软件定义网络技术遥测数据中心的网络和终端服务器的网络状态。

为了减少管控纠删码存储系统带来的开销,研究

学者提出了管控性能优化方案。Zhang 等[40]提出 RS6,减少 RAID-6 环境下执行行对角线奇偶校验(Row-Diagonal Parity,RDP)码扩展操作所消耗的编码计算和磁盘读写开销。Zhang 等[41]利用 SLAS(Sliding window, Lazy updates and movement Scheduling)方案来提升纠删码存储系统的可扩展性。Zhang 等[29]提出 McPod 方案,完成在 RAID-4 场景下的数据迁移操作,从而提升磁盘阵列存储系统的读写性能。通过将编码后的用户数据合理地放置在多个存储节点,Wu 等[42]提出 ERS(Elastic Reed-Solomon)码,减少纠删码编码参数转化过程带来的存储节点数据读写开销。Yao 等[43]提出 Stripe-Merge 方案,保证以较低的通信量完成多个窄条带合并为宽条带操作,实现在较小的存储开销的情况下保障数据的可靠性。

在此基础上,设计人员还部署了性能优化机制来提升纠删码存储系统的性能。Gong 等[44]提出数据修复过程中存储节点的选择机制,减少数据修复操作所消耗的时间。通过在大条带磁盘阵列存储系统场景下引入本地和跨机架编码机制,Hu 等[45]提出 EC-Wide 方案,减少数据修复传输开销。Tang 等[46]提出 UniDrive 数据读写方案,快速选择网络带宽资源较多的云存储节点下载数据。Plank 等[47]提出纠删码存储系统的开源实现方案,包括 Jerasure 库[48]和 Zfec 库[49],并给出了纠删码存储系统的通用教程[50]。由于存储系统状态管控方案针对不同应用场景下的性能瓶颈有较大的差异性,纠删码存储系统通常需要部署多种管控方案,以提升数据修复操作的性能。

3.2 数据放置方案

为了将数据合理地放置到多个磁盘,研究学者提出了多种数据放置方案。通过使用正交拉丁方矩阵将数据均衡地保存到多个磁盘,Zhang 等[20]提出 RAID+方案,将读写开销分配到各个磁盘,减少修复数据所消耗的时间。Yao 等[51]提出 Elastic-RAID 方案,引入多个副本来并行化磁盘读写操作。通过在磁盘阵列存储系统中保存额外的数据,Wan 等[52]提出 S2-RAID,减少数据修复过程读写的数据量。为了在保证存储空间利用率的前提下降低数据修复操作的性能,Wang 等[53]提出 OI-RAID,以并行化利用磁盘数据来减少数据修复操作所消耗的时间。

在较低存储空间开销的前提下,根据用户数据被读取的频率,Xia 等[54]将两种纠删码用于编码不同读取频率的用户数据,提升了纠删码存储系统执行降格读取操作和数据修复操作的性能。Wu 等[55]提出 WorkOut,将重构频率较高的数据保存到备用的磁盘阵列存储系统。Wan 等[56]提出 VDF-LRU 和 VDF-LFU,根据存储设备的失效概率来缓存数据,减少磁盘失效对数据读写的影响。

通过高效地访问存储节点,纠删码存储系统提高了数据修复操作的执行效率。Hou 等[57]提出一种通用的极大距离可分码(Maximum Distance Separable code,MDS)转化方法,在不改变更新的前提下减少数据修复操作消耗的网络带宽资源。Shen 等[58]设计了存储节点内存缓存区替换算法。Huang 等[59]提出局部修复码(Local Reconstruction Code,LRC),引入本地校验块来进一步减少数据修复操作的数据传输量。Wu 等[60]提出一种高效的 LRC 转化方法,平衡编码转化开销和修复开销。Wu 等[61]提出一种高效的 LRC 编码数据放置策略,以减少数据归并操作迁移过程中数据的传输开销。王意洁等[62]总结了分布式存储系统场景的容错技术。类似的工作还包括 Pyramid 码[63]、AR-TwoStep-ILP[64]、MiPiL[65]。

在此基础上,研究学者设计了适用于纠删码存储系统的数据存储机制以提升数据可靠性。Li 等[66]提出 OpenEC 编码库,使用有向无环图优化存储节点之间执行的数据编解码过程,以提升数据修复等操作的性能。在多个旋转条带应用场景下,Fu 等[67]提出两种搜索方案:均衡优先级(Balance Priority,BP)方案和搜索时间优先(Search Time Priority,STP)方案,进一步减少了数据修复过程中磁盘读写的开销。Hou 等[68]提出高效的跨机架精确修复方案,减少机架间的通信开销。Shen 等[69]提出寻道读写优化修复(Seek-efficient I/O Recovery,SIOR)算法,减少单节点修复时寻找磁盘的开销。Burihabwa 等[70]测试了当前主流的云存储服务安全机制及其性能开销。Shan 等[71]提出面向跨数据中心存储高效的数据备份方案,提高节点之间数据的传输效率。由于各种纠删码有着各自的编码规则,在设计数据放置方案时需要适应当前所使用的数据编码方案。

3.3 数据编码方案优化

为了保障数据的可靠性,大量纠删码被引入磁盘阵列存储系统,用于保存用户数据。Reed 等[1]利用 RS 码来提高用户数据的可靠性。由于 RS 码需要较大的计算开销,Plank 等[21]利用 Cauchy-RS 码来减少数据编码开销。为了进一步减少磁盘阵列存储系

统的编码开销,Blaum 等[12]提出 EVENODD 码,提升 RAID-6 场景下的数据编码性能。此外,Blaum 等[72]利用 Blaum-Roth 码来减少编码过程中异或操作的次数。Fu 等[73]提出 D-Code,提升数据读取性能。Jin 等[74]提出 P-Code 码,在保证存储效率的同时减少数据修复和更新操作时的编解码开销。Huang 等[75]提出星形(STAR)码来支持三磁盘容错。Blaum 等[76]提出 EBR 方案提升 Blaum-Roth 码数据编解码的效率。类似的编码方案还包括 RDP 码[77]、Liberation 码[78]和 Liber8tion 码[79]。

在此基础上,研究学者通过研究多种编码机制来提升磁盘阵列存储系统的数据修复性能和容错能力。Zhang 等[80]提出三独立磁盘奇偶校验(Triple Independent Parity,TIP)码,减少三磁盘容错情况下的数据更新编码开销。为了应对磁盘阵列存储系统中存在大量的扇区操作,Plank 等[81]利用 SD (Sector-Disk)码来保障数据的可靠性。由于 SD 码存在扇区容错数量限制,Li 等[82]进一步提出阶梯(STAIR)码来实现多个扇区出现故障时的数据修复操作。Plank 等[83]开源实现了多种 RAID-6 编码方案,并比较了 RAID-6 编码过程中数据编解码操作的性能。Wu 等[84]利用 Expanded-Blaum-Roth 码来提升 Blaum-Roth 码的编解码效率。Khan 等[85]使用最短路径算法分析了各种磁盘阵列纠删码修复最少消耗的磁盘读写开销。类似的编码方案还包括 TIP 码[80]、X-Code[86]、HoVer 码[87]、STAR+码[88]。

为了减少存储节点之间传输的数据量,研究学者设计了高效的数据编码机制。Dimakis 等[23]提出再生码,减少存储节点之间传输的数据量,并证明了该编码方案能通过传输最小的数据量完成单个存储节点的修复操作。在此基础上,Shum[89]提出合作再生码(Cooperative Regenerating Code,CRC),进一步减少数据修复过程中传输的数据量,并给出了合作数据修复过程中需要传输的最小数据量[90]。Hou 等[91]提出机架感知再生码(Rack-aware Regenerating Codes,RRC),减少跨机架间大网络传输开销。在此基础上,Hou 等[92]提出通用机架感知再生码(Generalized Rack-aware Regenerating Codes,GRRC),同时减少在节点失效和机架失效情况下修复过程的数据传输量。杨松霖等[93]综述了当前主要的数据修复方法,并对这些方法的实现机制进行了分析。

云际存储系统的数据编码方案除了需要保障存储服务的可用性外,还需要考虑数据的安全性和存储空间的利用率。为了保障存储数据的安全性,Resch 等[94]提出 AONT-RS (All-Or-Nothing Transform with Reed-Solomon)算法。由于 AONT-RS 使用随机数来加密用户数据,造成编码后数据难以执行重删操作,因此 Li 等[95]使用 CAONT-RS (Convergent AONT-RS)算法优化了重删操作的执行效率,开发了 CDStore 云际存储系统来减少存储空间的消耗量[96]。

研究学者还提出多种云际存储系统编码性能优化方案。Shen 等[97]提出基于流水线的数据写入方案,以充分利用用户端的计算能力和传输资源。通过引入再生码机制,Chen 等[14]提出 NCCloud 方案,减少执行数据修复操作时云存储节点之间传输的数据量。朱良杰等[98]总结了当前云际存储系统的主要编码算法和性能优化方案。类似的研究方案还包括 NCCS[9]、CloudS[99]、RACS[100]。

表 4 展示了常见的纠删码存储系统数据编码算法的主要应用场景和性能优化目标。由于各种编码算法适用的应用场景和性能优化指标有较大的差异,因此需要在纠删码存储系统应用场景部署相应的数据编码方案,以保障纠删码存储系统的可靠性。

表 4　常见的数据编码算法主要应用场景和性能优化目标

Table 4　Main application scenarios and performance optimization objectives of common data coding algorithms

编码算法 Coding algorithm	主要应用场景 Main application scenarios	性能优化目标 Performance optimization objectives	优势 Advantages	局限性 Limitations
EVENODD[12]	RAID systems	High coding performance	Low coding computation overhead	Support single coding parameters
RDP[77]	RAID systems	High coding performance	Low coding computation overhead	Support single coding parameters
Liberation[78]	RAID systems	High coding performance	Low coding computation overhead	Support single coding parameters
Liber8tion[79]	RAID systems	High coding performance	Low coding computation overhead	Support single coding parameters

续表

Continued table

编码算法 Coding algorithm	主要应用场景 Main application scenarios	性能优化目标 Performance optimization objectives	优势 Advantages	局限性 Limitations
TIP[80]	RAID systems	Support three-disk error tolerance	Low coding computation overhead	Support single coding parameters
Blaum-Roth[72]	RAID systems	High coding performance	Low coding computation overhead	Support single coding parameters
D-Code[73]	RAID systems	High disk I/O performance	High data read and degraded read performance	Support single coding parameters
P-Code[74]	RAID systems	High recovery and update performance	Low coding computation overhead	Support single coding parameters
HoVer[87]	RAID systems	Support four-disk error tolerance, high performance of data recovery process	Low coding computation overhead, high date reliability	Non-MDS code
Pyramid[63]	RAID systems	High data recovery performance	High performance of data recovery	Non-MDS code
STAR[75]	RAID systems	Tolerate three-disk error	Low coding computation overhead	Support single coding parameters
STAR +[88]	RAID systems	High coding/decoding performance	Low coding computation overhead	Support single coding parameters
MDR[22]	RAID systems	Low disk I/O overhead	High single-disk recovery performance	Support single coding parameters
SD-Code[81]	RAID systems	Tolerate sector error	Tolerate up to three sectors error	Support single coding parameters
STAIR[82]	RAID systems	Tolerate sector error	Tolerate any number of sector error	High coding computation overhead
RS[1]	Distributed storage system	High generality	Support multiple coding parameters	High coding computation overhead
Cauchy-RS[21]	Distributed storage system	High generality	Support multiple coding parameters	High coding computation overhead
RC[23]	Distributed storage system	Low consumption of network bandwidth resource	Low consumption of network bandwidth resource	Only support functional recovery
CRC[89]	Distributed storage system	Low consumption of network bandwidth resource	Low consumption of network bandwidth resource	Only support functional recovery
LRC[59]	Distributed storage system	High performance of data recovery	Low consumption of network bandwidth resource	Non-MDS code
RRC[91]	Distributed storage system	High performance of data recovery process across multiple racks	High performance of date transmission efficiency across multiple racks	Increase the data transmission within the racks
GRRC[92]	Distributed storage system	High performance of data recovery across multiple racks	High performance of date transmission efficiency across multiple racks	Increase the data transmission within the racks
NCCS[9]	Cloud-of-clouds storage systems	High I/O performance of mobile user device	Reduce the write overhead from the mobile terminal	Increase the transmission overhead between clouds
NCCloud[14]	Cloud-of-clouds storage systems	High performance of data recovery	Low consumption of network bandwidth resource	Only support functional recovery
AONT-RS[94]	Cloud-of-clouds storage systems	High security	Ensure the data security of cloud side	Cannot efficiently utilize the storage space
CAONT-RS[95]	Cloud-of-clouds storage systems	High security, support deduplication	High utilization of storage space	Cannot ensure the security of user devices
CloudS[99]	Cloud-of-clouds storage systems	High security and coding/decoding performance	Support hierarchical data security	Non-MDS code
RACS[100]	Cloud-of-clouds storage systems	High performance of the parallel I/O operations	Provide the RAID storage service in cloud side	High overhead of synchronous operations

3.4 数据修复过程性能优化

研究学者提出编码调度算法来减少磁盘阵列存储系统数据编码操作带来的计算开销。根据数据字内容(Data Words Guided,DWG),Luo 等[101]提出数据字驱动编码调度方案(DWG XOR-Scheduling),用于减少数据编解码过程中的编解码计算。在此基础上,Luo 等[102]提出基于图搜索的编码计算方案来减少纠删码存储系统的编解码开销。Huang 等[103]证明了数据编码过程计算调度优化问题是不确定多项式(Non-determinstic Polynomial,NP)问题,并提出了 Subex 算法提高编码计算效率。Hafner 等[104]提出特定编码混合构建优化(Code Specific Hybrid Reconstruction Optimization,CSHR)方案,用于减少磁盘阵列存储系统矩阵阵列编码过程的计算开销。在此基础上,Plank 等[105]提出 Uber-CSHR 方案来进一步提高修复失效数据的编码计算效率。

研究学者提出多种编码计算调度算法来减少磁盘阵列存储系统执行数据修复操作的磁盘读写开销。Xiang 等[106]提出 RDOR 修复优化方案,首次减少了 RDP 编码数据修复操作过程中的磁盘读写开销。Khan 等[107]利用修复编码图搜索方案和 Rotate RS 码来减少执行 RS 码数据修复过程中的磁盘开销。傅颖勋等[108]总结了当前单磁盘失效情况下的数据修复方案。类似的性能优化工作还包括 MDRR[109]、PDRS[110]、CaCo[111]。

由于分布式存储系统中存储节点之间的网络资源存在较大的异构性,因此研究学者通过优化存储节点之间的传输方案以减少数据修复时间。Li 等[112]利用基于树形拓扑的数据修复算法(RCTREE)[4]来减少数据修复操作所消耗的时间。在此基础上,Pei 等[113]提出 CTREE 方案,修复多存储节点失效场景下的数据。Wang 等[114]提出 FastRC 方案来优化数据修复操作的数据传输方案。由于传输的数据量不足,以上方案并不能保证修复过程中数据的完整性。

Shah 等[115]提出弹性再生码(Flexible regenerating code),通过充分利用存储节点之间的网络带宽资源,减少数据修复消耗的时间。通过引入弹性修复机制,Wang 等[6]提出弹性树形修复(Flexible Tree Regeneration,FTR)方案,保障修复后数据的完整性。Huang 等[28]提出 PUSH 方案,将编码计算和网络传输开销分散到多个存储节点以提升数据修复的效率。由于存储节点之间网络带宽具有较大的波动性,Shen 等[7]利用基于网络带宽感知的数据修复方案来减少数据修复操作所消耗的时间。Shen 等[116]推导出合作数据修复过程中传输数据量的下限,并提出使用弹性合作修复码(Flexible Cooperative Regenerating Code,FCRC)来决策存储节点之间传输的数据量,减少合作修复过程消耗的时间。Shen 等[117]提出异构网络感知的合作修复(Heterogeneity-aware Cooperative Regeneration,HCR)路由规划方案,完成合作修复过程中的数据传输路由生成任务。

在此基础上,研究学者结合具体应用场景进一步提高了数据修复方案的执行效率。Plank 等[118]将低密度奇偶校验(Low-Density Parity-Check,LDPC)码部署于广域网存储系统,用于减少数据传输开销以保障数据的可靠性。Mitra 等[119]提出局部并行化修复(Partial-Parallel-Repair,PPR)方案,并行化存储节点之间的数据传输操作,以利用存储节点之间的带宽资源。Rashmi 等[120]提出一种基于生成矩阵的最小存储空间开销修复(Product Matrix-Minimum Storage Regeneration,PM-MSR)码[121],称为 PM-RBT,用于同时减少数据修复过程中的数据传输开销和磁盘读写开销。Yao 等[122]提出 PivotRepair,构建并行化传输方案,以充分利用节点之间的网络资源。钟凤艳等[123]总结了存储节点异构情况下失效数据修复的方案。

表 5 总结了当前数据修复过程中主要的性能优化方法的应用场景和优化目标。由于纠删码存储系统被部署到多个应用场景,当前数据修复过程中通常使用编码计算调度[102]、数据传输路径优化[4]、并行化数据传输操作[119]等多种性能优化机制来提升数据修复操作的性能。

由表 5 可知,使用单一的性能优化方案难以满足纠删码存储系统对数据性能的需求,因此需要综合部署多种数据修复性能优化方案,保障用户数据的可靠性。此外,随着在线应用存储数据量的增加,纠删码存储系统需要更多的存储节点用于保存用户数据。数据修复方案也应充分使用多个存储节点的计算、存储和网络资源,以提高纠删码存储系统数据修复操作的执行效率。

表 5　数据修复过程性能优化相关技术的应用场景和优化目标

Table 5　Application scenarios and optimization objectives of performance optimization related technologies during data recovery process

优化方法 Optimization approaches	应用场景 Application scenarios	优化目标 Optimization targets	优势 Advantages	局限性 Limitations
XOR-Scheduling[102]	RAID systems	High coding performance	Low coding computation overhead	Not consider disk I/O overhead
CSHR[104]	RAID systems	High coding performance	Low coding computation overhead	Not consider disk I/O overhead
Uber-CSHR[105]	RAID systems	High coding performance	Low coding computation overhead	Not consider disk I/O overhead
Subex[103]	RAID systems	High coding performance	Low coding computation overhead	Not consider disk I/O overhead
RDOR[106]	RAID systems	High data recovery performance	Low disk I/O overhead	Only support RDP code
MDRR[109]	RAID systems	High data recovery performance	Low disk I/O overhead	Only support X-Code
CaCo[111]	RAID systems	High coding performance	Low coding computation overhead	Only support Cauchy-RS code
RCTREE[4]	Distributed storage system	Efficient tree-topology date recovery scheme	Tree topology recovery scheme	Cannot ensure data integrity
FTR[6]	Distributed storage system	Efficient flexible tree-topology recovery scheme	Flexible tree-topology recovery scheme	Only support single-node flexible recovery
PUSH[28]	Distributed storage system	Efficient parallel data transmission scheme	Fully utilize computation resource of storage nodes	Need multi-node collaborative coding data
CTREE[113]	Distributed storage system	Cooperative recovery scheme	Tree-topology cooperative recovery scheme	Cannot ensure the data integrity
FastRC[114]	Distributed storage system	Tree-topology recovery scheme	Tree-topology recovery scheme	Cannot determine data volume of transmission operations
FRC[115]	Distributed storage system	Efficient flexible date recovery scheme	Flexible cooperative recovery for single storage node	Cannot construct tree-topology recovery scheme
FCRC[116]	Distributed storage system	Efficient flexible cooperative recovery scheme	Flexible cooperative recovery for multiple storage nodes	Cannot construct the tree-topology recovery scheme
HCR[117]	Distributed storage system	Efficient cooperative recovery scheme	Support construct multi-node routing scheme	Cannot determine the transmission amount
PPR[119]	Distributed storage system	Efficient parallel data transmission scheme	Parallel data transmission operations	Complex network communication control protocol
PM-RBT[120]	Distributed storage system	High disk I/O and transmission performance	Low disk I/O and transmission overhead	High coding computation overhead
PivotRepair[122]	Distributed storage system	Efficient parallel data transmission scheme	Fully utilize the network bandwidth resources	Only support single-node tree topology recovery

3.5　数据修复过程性能优化技术的部署要点

为了高效地修复失效节点存储的数据,纠删码存储系统需要部署合适的数据编码算法和对应的数据修复性能优化技术方案。在部署数据修复方案过程中,设计人员需要注意以下要点:首先,由于纠删码被部署到多种类型的存储系统中,数据修复性能优化方案通常仅针对某个特定的应用场景,且其优化性能指标有较大的差异,因此纠删码存储系统设计人员需要根据应用场景和优化目标选择合适的性能优化方案。其次,为了保障数据修复操作的性能,部署数据修复方案过程中通常需要选择多项相关的性能优化技术,并根据应用场景的性能需求来调整性能优化方案的实现机制。表 6 展示了各种数据修复过程性能优化关键技术的部署要点、优势和局限性。

表 6　数据修复过程性能优化关键技术的部署要点、优势和局限性

Table 6　Key deployment points, advantages and limitations of key technologies for data recovery process performance optimization

关键技术 Key technologies	部署要点 Deployment principles	优势 Advantages	局限性 Limitations
State management of storage systems	Get the current state of storage nodes and migrate the data of storage nodes with high failure possibility	Migrate the data can reduce the possibility of data recovery and improve data reliability of storage systems	When the failures of storage nodes occur, erasure-coded storage systems still need to perform data recovery process
Optimization of data placement schemes	Storage systems can obtain current state of storage nodes and network, and select the date storage nodes to place the user data	Storage systems can fully utilize the network bandwidth between storage nodes to improve the performance of data recovery process	Storage systems can hardly reduce data volume of disk I/O operations and network transmission operations
Optimization of coding schemes	According to the performance requirement of data recovery process in application scenarios, the developer can select the coding scheme to improve the performance of data recovery process	Coding scheme can reduce the performance overhead to satisfy the performance requirements of application scenarios	Storage systems can hardly adjust the recovery scheme, according to the current state of erasure-coded storage systems
Optimization of data recovery process	According to the storage node and network conditions of the application scenario, select the optimization scheme to improve the performance of data recovery process	According to the status change of the erasure-coded storage system, dynamically adjust the execution process of data recovery operation	Relying on data coding and placement scheme, it is difficult to fundamentally improve the execution efficiency of data recovery process

4　展望

4.1　学术研究领域

虽然已有较多研究工作致力于提升数据修复操作的性能，但是数据修复操作的性能优化研究在学术研究领域仍存在以下值得研究的内容。

第一，存储设备管控技术缺乏逻辑推理能力。为了提高预测存储设备状态的准确率，当前设备状态预测通常大量使用统计机器学习算法，如深度学习算法[20]。由于统计机器学习算法难以准确判断存储设备故障的原因，运维人员仍需要分析故障原因，并调整相应的系统配置。因此，当前的存储设备管控技术亟需提升逻辑推理能力，以帮助纠删码存储系统快速修复失效数据。

第二，动态网络传输性能优化。虽然在固定的网络拓扑情况下能取得较好的数据修复效果[6]，但是当前的数据修复方案仍难以应对动态网络应用场景。由于分布式存储系统中存储节点的网络存储存在较大的动态性[7]，因此数据修复方案应进一步考虑纠删码存储系统中在可用网络带宽波动情况下的数据修复性能优化相关技术，以提升数据修复操作对动态网络的适应性。

第三，全流程优化数据修复操作执行过程。当前性能优化机制专注于保障数据修复操作某一阶段的执行效率。然而，纠删码存储系统通常需要综合多个阶段的性能优化技术，才能保障数据修复操作性能[15]。因此纠删码存储系统亟需面向全流程的数据修复操作性能优化方案，保障纠删码存储系统保存数据的可靠性。

4.2　工业应用领域

为了保障纠删码存储系统实际应用场景下用户数据的可靠性，数据修复方案性能优化在工业应用领域存在以下值得研究的内容。

第一，提升数据修复方案的场景适应性。当前数据修复方案主要针对特定的应用场景。然而，在实际工业领域中，纠删码存储系统通常需要承接多种不同的在线应用负载[11]。因此，设计适用于多种应用场景的数据修复性能优化方案仍是值得研究的课题。

第二，满足大规模存储场景服务的质量需求。由于存储数据量的增加，传统数据修复方案已经难以满足快速修复失效节点数据的需求[20]。为了满足大规模存储系统的数据可靠性需求，数据修复机制需要并行化利用存储节点的计算和网络资源，提升数据修复操作的性能。

第三，提升数据修复算法的智能化程度。虽然当前磁盘故障预测等领域已应用一些人工智能算法，但是许多数据修复技术仍使用传统启发式性能优化方案，难以应对当前日益复杂的应用场景[122]。针对各种在线应用场景的特点，合理使用人工智能等新的性能优化方法将成为提高数据修复操作执行效率的重要途径。

参考文献

[1]　REED I S, SOLOMON G. Polynomial codes over certain

finite fields [J]. Journal of the Society for Industrial and Applied Mathematics,1960,8(2):300-304.

[2] 张耀,储佳佳,翁楚良. 纠删码存储系统数据更新方法研究综述[J]. 计算机研究与发展,2020,57(11):2419-2431.

[3] SATHIAMOORTHY M,ASTERIS M,PAPAILIOPOULOS D,et al. XORing elephants:novel erasure codes for big data [C]//Proceedings of 39th International Conference on Very Large Data Bases. Riva del Garda,Trento,Italy:[s. n.],2013:325-336.

[4] LI J,YANG S,WANG X,et al. Tree-structured data regeneration in distributed storage systems with regenerating codes [C]//Proceedings of IEEE Conference on Computer Communication (INFOCOM 10). San Diego,CA,USA:IEEE,2010:1-9.

[5] REINSEL D,武连峰,GANTZ J F,等. IDC:2025 年中国将拥有全球最大的数据圈(白皮书) [EB/OL]. (2019-01) [2022 - 05 - 04]. https://www. docin. com/p - 2182394103. html.

[6] WANG Y,WEI D S,YIN X R,et al. Heterogeneity aware data regeneration in distributed storage systems [C]//Proceedings of IEEE Conference on Computer Communications (INFOCOM 14). Toronto,ON,Canada:IEEE,2014:1878-1886.

[7] SHEN J J,GU J Z,ZHOU Y F,et al. Bandwidth-aware delayed repair in distributed storage systems [C]//Proceedings of 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS 16). Beijing,China:IEEE,2016:1-10.

[8] PLANK J S. Erasure codes for storage systems:a brief primer [J]. Login:the Magazine of USENIX & SAGE,2013,38(6):44-50.

[9] SHEN J,LI Y,ZHOU Y,et al. Mobile cloud-of-clouds storage made efficient:a network coding based approach [C]//Proceedings of 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS 18). Salvador,Brazil:IEEE,2018:72-82.

[10] MENON J,MATTSON D. Distributed sparing in disk arrays [C]//Thirty-Seventh IEEE Computer Society International Conference. San Francisco, CA, USA:IEEE,1992:410-421.

[11] SHEN J J,ZHANG K,GU J Z,et al. Efficient scheduling for multi-block updates in erasure coding based storage systems [J]. IEEE Transactions on Computers,2017,67(4):573-581.

[12] BLAUM M,BRADY J,BRUCK J,et al. EVENODD:an efficient scheme for tolerating double disk failures in raid architectures [J]. IEEE Transactions on Computers,1995,44(2):192-202.

[13] BESSANI A,CORREIA M,QUARESMA B,et al. DepSky:dependable and secure storage in a cloud-of-clouds [J]. ACM Transactions on Storage,2013,9(4):1-33.

[14] CHEN H C H,HU Y C,LEE P P C,et al. NCCloud:a network-coding based storage system in a cloud-of-clouds [J]. IEEE Transactions on Computers,2014,63(1):31-44 .

[15] 沈佳杰. 纠删码存储系统构建及性能优化关键技术研究[D]. 上海:复旦大学,2020.

[16] 王艳. 基于纠删码的分布式存储系统的数据修复性能研究[D]. 上海:复旦大学,2014.

[17] HAN S J,LEE P P C,SHEN Z R,et al. Toward adaptive disk failure prediction via stream mining [C]//Proceedings of 2020 IEEE 40th International Conference on Distributed Computing Systems. Singapore:IEEE,2020:628-638.

[18] 沈佳杰,朱良杰,向望,等. 大规模高校纠删码键值存储读写负载均衡研究[J]. 深圳大学学报(理工版),2020,37(Z1):175-183.

[19] LIU K Y,PENG J,WANG J R,et al. A learning-based data placement framework for low latency in data center networks [J]. IEEE Transactions on Cloud Computing,2019,10(1):146-157.

[20] ZHANG G Y,HUANG Z C,MA X,et al. RAID+ :deterministic and balanced data distribution for large disk enclosures [C]//Proceedings of the 16th USENIX Conference on File and Storage Technologies (FAST 18). Berkeley, CA, USA: USENIX Association, 2018:279-293.

[21] PLANK J S,XU L H. Optimizing Cauchy Reed-Solomon codes for fault-tolerant network storage applications [C]//Proceedings of the 5th IEEE International Symposium on Network Computing and Applications. Washington,DC,USA:IEEE Computer Society,2006:173-180.

[22] WANG Y,YIN X R,WANG X. MDR codes:a new class of RAID-6 codes with optimal rebuilding and encoding [J]. IEEE Journal on Selected Areas in Communications,2014,32(5):1008-1018.

[23] DIMAKIS A G,GODFREY P B,WU Y,et al. Network coding for distributed storage systems [J]. IEEE Transactions on Information Theory, 2010, 56 (9):4539-4551.

[24] DIMAKIS A G,RAMCHANDRAN K,WU Y N,et al.

A survey on network codes for distributed storage [J]. Proceedings of the IEEE,2011,99(3):476-489.

[25] JIEKAK S,KERMARREC A M,SCOUARNEC N L, et al. Regenerating codes: a system perspective [J]. ACM SIGOPS Operating Systems Review, 2013, 47(2):23-32.

[26] ZHENG W M,ZHANG G Y. FastScale:accelerate RAID scaling by minimizing data migration [C]//Proceedings of the 9th USENIX Conference on File and Storage Technologies. Berkeley, CA, USA: USENIX Association,2011:1-13.

[27] 维基百科. S. M. A. R. T. 介绍[EB/OL]. [2022-05-04]. https://zh. wikipedia. org/wiki/S. M. A. R. T.

[28] HUANG J Z,XIE C S,LIANG X H,et al. PUSH:a pipelined reconstruction I/O for erasure-coded storage clusters [J]. IEEE Transactions on Parallel and Distributed Systems,2015,26(2):516-526.

[29] ZHANG G Y,WANG J,LI K Q,et al. Redistribute data to regain load balance during RAID-4 scaling [J]. IEEE Transactions on Parallel and Distributed Systems,2014,26(1):219-229.

[30] LU S D,LUO B,PATEL P,et al. Making disk failure predictions SMARTer! [C]//Proceedings of the 18th USENIX Conference on File and Storage Technologies. Berkeley,CA, USA: USENIX Association, 2020: 151-168.

[31] XU F,HAN S J,LEE P P C,et al. General feature selection for failure prediction in large-scale SSD deployment [C]//Proceedings of the 51st IEEE/IFIP International Conference on Dependable Systems and Networks. Taipei,Taiwan,China:IEEE,2021:263-270.

[32] HAN S J,LEE P P C,XU F,et al. An in-depth study of correlated failures in production SSD-based data centers [C]//Proceedings of the 19th USENIX Conference on File and Storage Technologies. Berkeley,CA,USA: USENIX Association,2021:1-13.

[33] LIN Q W,HSIEH K,DANG Y N,et al. Predicting node failure in cloud service systems [C]//Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York: ACM,2018:480-490.

[34] LI Y G,JIANG Z M,LI H,et al. Predicting node failures in an ultra-large-scale cloud computing platform: an ALOps solution [J]. ACM Transactions on Software Engineering and Methodology,2020,29(2):1-24.

[35] LIN S Y,GONG G W,SHEN Z R,et al. Boosting full-node repair in erasure-coded storage [C]//Proceedings of the 2021 USENIX Annual Technical Conference. Berkeley,CA, USA: USENIX Association, 2021: 641-655.

[36] LI R,CHENG Z N,LEE P P C,et al. Automated intelligent healing in cloud-scale data centers [C]//Proceedings of the 40th International Symposium on Reliable Distributed Systems. Chicago, IL, USA: IEEE, 2021:244-253.

[37] GUO J,CHANG Z H,WANG S,et al. Who limits the resource efficiency of my datacenter:an analysis of Alibaba datacenter traces [C]//Proceedings of 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS 19). Phoenix, AZ, USA: IEEE, 2019:1-10.

[38] CHENG Z N,TANG L,HUANG Q. Enabling low-redundancy proactive fault tolerance for stream machine learning via erasure coding [C]//Proceedings of the 40th International Symposium on Reliable Distributed Systems. Chicago,IL,USA:IEEE,2021:99-108.

[39] HUANG Q,SUN H,LEE P P C,et al. OmniMon:re-architecting network telemetry with resource efficiency and full accuracy [C]//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications,Technologies,Architectures,and Protocols for Computer Communication (SIGCOMM 20). New York: ACM, 2020: 404-421.

[40] ZHANG G Y,LI K Q,WANG J,et al. Accelerate RDP raid-6 scaling by reducing disk I/Os and XOR operations [J]. IEEE Transactions on Computers, 2013, 64(1):32-44.

[41] ZHANG G Y,SHU J W,XUE W,et al. SLAS:an efficient approach to scaling round-robin striped volumes [J]. ACM Transactions on Storage,2007,3(1):1-39.

[42] WU S,SHEN Z R,LEE P P C. Enabling I/O-efficient redundancy transitioning in erasure-coded KV stores via elastic Reed-Solomon codes [C]//Proceedings of 2020 International Symposium on Reliable Distributed Systems (SRDS 20). Shanghai,China:IEEE,2020:246-255.

[43] YAO Q R,HU Y C,CHENG L F,et al. StripeMerge: efficient wide-stripe generation for large-scale erasure-coded storage [C]//Proceedings of the 41st IEEE International Conference on Distributed Computing Systems (ICDCS 21). Washington,DC,USA:IEEE,2021: 483-493.

[44] GONG Q Y,WANG J Q,WEI D S,et al. Optimal node selection for data regeneration in heterogeneous distributed storage systems [C]//Proceedings of the 44th International Conference on Parallel Processing (ICPP 15). Beijing,China:IEEE,2015:390-399.

[45] HU Y,CHENG L,YAO Q,et al. Exploiting combined locality for wide-stripe erasure coding in distributed storage [C]//Proceedings of the 19th USENIX Conference on File and Storage Technologies. Berkeley, CA,USA:USENIX Association,2021:233-248.

[46] TANG H W,LIU F M,SHEN G B,et al. UniDrive: synergize multiple consumer cloud storage services [C]//Proceedings of the 16th Annual Middleware Conference. New York:ACM,2015:137-148.

[47] PLANK J S,LUO J Q,SCHUMAN C D,et al. A performance evaluation and examination of open-source erasure coding libraries for storage [C]//Proceedings of the 4th USENIX Conference on File and Storage Technologies (FAST 09). Berkeley,CA,USA:USENIX Association,2009:253-265.

[48] PLANK J S,SIMMERMAN S,SCHUMAN C D. Jerasure:a library in C/C++ facilitating erasure coding for storage applications-version 1.2 [Z/OL]. [2022-05-04]. http://www.cs.utk.edu/~plank/plank/papers/CS-08-627.html.

[49] WILCOX-O'HEARN Z. Zfec,open source code distribution [Z/OL]. [2022-05-04]. http://pypi.org/project/zfec.

[50] PLANK J S,HUANG C. Tutorial:erasure coding for storage systems [C]//Proceedings of the 11th USENIX Conference on File and Storage Technologies. Berkeley,CA,USA:USENIX Association,2005:1-74.

[51] YAO J,JIANG H,CAO Q,et al. Elastic-RAID:a new architecture for improved availability of parity-based RAIDs by elastic mirroring [J]. IEEE Transactions on Parallel and Distributed Systems,2016,27(4):1044-1056.

[52] WAN J G,WANG J B,YANG Q,et al. S2-RAID:a new raid architecture for fast data recovery [C]//Proceedings of IEEE 26th Symposium on Mass Storage Systems and Technologies. Incline Village,NV,USA: IEEE,2010:1-9.

[53] WANG N,XU Y L,LI Y K,et al. OI-RAID:a two-layer RAID architecture towardsfast recovery and high reliability [C]//Proceedings of the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Washington,DC,USA:IEEE Computer Society,2016:61-72.

[54] XIA M Y,SAXENA M,BLAUM M,et al. A tale of two erasure codes in HDFS [C]//Proceedings of the 13th USENIX Conference on File and Storage Technologies. Berkeley,CA,USA:USENIX Association,2015: 213-226.

[55] WU S Z,JIANG H,ENG D,et al. WorkOut:I/O workload outsourcing for boosting RAID reconstruction performance [C]//Proceedings of the 7th USENIX Conference on File and Storage Technologies. Berkeley, CA,USA:USENIX Association,2009:239-252.

[56] WAN S G,HE X B,HUANG J Z,et al. An efficient penalty-aware cache to improve the performance of parity-based disk arrays under faulty conditions [J]. IEEE Transactions on Parallel and Distributed Systems,2013,24(8):1500-1513.

[57] HOU H X,LEE P P C,HAN Y S. Toward optimality in both repair and update via generic MDS code transformation [C]//Proceedings of 2020 IEEE International Symposium on Information Theory. Los Angeles, CA,USA:IEEE,2020:560-565.

[58] SHEN J J,LI Y,SHENG G W,et al. Efficient memory caching for erasure coding based key-value storage systems [C]//Proceedings of the 6th CCF Conference on Big Data. Xi'an,Shaanxi,China:CCF,2018:512-539.

[59] HUANG C,SIMITCI H,XU Y K,et al. Erasure coding in windows azure storage [C]//Proceedings of the 2012 USENIX Conference on Annual Technical Conference. Berkeley, CA, USA: USENIX Association, 2012:15-26.

[60] WU S,SHEN Z R,LEE P P C,et al. Optimal repair-scaling trade-off in locally repairable codes:analysis and evaluation [J]. IEEE Transactions on Parallel and Distributed Systems,2022,33(1):56-69.

[61] WU S,DU Q P,LEE P P C,et al. Optimal data placement for stripe merging in locally repairable codes [C]//Proceedings of IEEE International Conference on Computer Communications. London,UK:IEEE,2022: 1-10.

[62] 王意洁,许方亮,裴晓强. 分布式存储中的纠删码容错技术研究[J]. 计算机学报,2017,40(1):236-255.

[63] HUANG C,CHEN M H,LI J. Pyramid codes:flexible schemes to trade space for access efficiency in reliable data storage systems [C]//Proceedings of the 6th IEEE International Symposium on Network Computing and Applications. Washington, DC, USA: IEEE Computer Society,2007:79-86.

[64] YAO J J,LU P,GONG L,et al. On fast and coordinated data backup in geo-distributed optical inter-datacenter networks [J]. Journal of Lightwave Technology,2015,33(14):3005-3015.

[65] ZHANG G Y,ZHENG W M,LI K Q. Rethinking RAID-5 data layout for better scalability [J]. IEEE Transactions on Computers,2013,63(11):2816-2828.

[66] LI X,LI R,LEE P P C,et al. OpenEC:toward unified and configurable erasure coding management in distributed storage systems [C]//Proceedings of the 17th USENIX Conference on File and Storage Technologies. Berkeley,CA,USA:USENIX Association,2019:331-344.

[67] FU Y X,SHU J W,SHEN Z R,et al. Reconsidering single disk failure recovery for erasure coded storage systems:optimizing load balancing in stack-level [J]. IEEE Transactions on Parallel and Distributed Systems,2016,27(5):1457-1469.

[68] HOU H X,LEE P P C,HAN Y S. Minimum storage rack-aware regenerating codes with exact repair and small sub-packetization [C]//Proceedings of 2020 IEEE International Symposium on Information Theory (ISIT 20). Los Angeles,CA,USA:IEEE,2020:554-559.

[69] SHEN Z R,SHU J W,LEE P P C,et al. Seek-efficient I/O optimization in single failure recovery for XOR-Coded storage systems [J]. IEEE Transactions on Parallel and Distributed Systems,2017,28(3):877-890.

[70] BURIHABWA D,PONTES R,FELBER P,et al. On the cost of safe storage for public clouds:an experimental evaluation [C]//Proceedings of the 35th IEEE Symposium on Reliable Distribute Systems (SRDS 16). Washington,DC,USA:IEEE Computer Society,2016:157-166.

[71] SHAN Y,CHEN K,GONG T,et al. Geometric partitioning:explore the boundary of optimal erasure code repair [C]//Proceedings of ACM SIGOPS 28th Symposium on Operating Systems Principles (SOSP 21). New York:ACM,2021:457-471.

[72] BLAUM M,ROTH R M. On lowest density MDS codes [J]. IEEE Transactions on Information Theory,1999,45(1):46-59.

[73] FU Y X,SHU J W. D-Code:an efficient RAID-6 code to optimize I/O loads and read performance [C]//Proceedings of 2015 IEEE International Parallel and Distributed Processing Symposium (IPDPS 15). Washington,DC,USA:IEEE Computer Society,2015:603-612.

[74] JIN C,JIANG H,FENG D,et al. P-Code:a new RAID-6 code with optimal properties [C]//Proceedings of the 23rd International Conference on Supercomputing. New York:ACM,2009:360-369.

[75] HUANG C,XU L X. STAR:an efficient coding scheme for correcting triple storage node failures [J]. IEEE Transactions on Computers,2008,57(7):889-901.

[76] BLAUM M,DEENADHAYALAN V,HETZLER S. Expanded blaum-roth codes with efficient encoding and decoding algorithms [J]. IEEE Communications Letters,2019,23(6):954-957.

[77] CORBETT P,ENGLISH B,GOEL A,et al. Row-diagonal parity for double disk failure correction [C]//Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST 04). Berkeley,CA,USA:USENIX Association,2004:1-14.

[78] PLANK J S. The RAID-6 liberation codes [C]//Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST 08). Berkeley,CA,USA:USENIX Association,2008:26-29.

[79] PLANK J S. The RAID-6 liber8tion code [J]. The International Journal of High Performance Computing Applications,2009,23(3):242-251.

[80] ZHANG Y Z,WU C T,LI J,et al. TIP-code:a three independent parity code to tolerate triple disk failures with optimal update complexity [C]//Proceedings of 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Washington,DC,USA:IEEE Computer Society,2015:136-147.

[81] PLANK J S,BLAUM M,HAFNER J L. SD codes:erasure codes designed for how storage systems really fail [C]//Proceedings of the 11th USENIX Conference on File and Storage Technologies. Berkeley,CA,USA:USENIX Association,2013:95-104.

[82] LI M Q,LEE P P C. STAIR codes:a general family of erasure codes for tolerating device and sector failures [J]. ACM Transactions on Storage,2014,10(4):1-30.

[83] PLANK J S,BUCHSBAUM A L,VANDER ZANDEN B T. Minimum density RAID-6 codes [J]. ACM Transactions on Storage,2011,6(4):1-22.

[84] WU Y,HOU H X,HAN Y S,et al. Generalized Expanded-Blaum-Roth codes and their efficient encoding/decoding [C]//Proceedings of IEEE Global Communications Conference. New York:IEEE,2020:1-6.

[85] KHAN O,BURNS R,PLANK J. In search of I/O-optimal recovery from disk failures [C]// Proceedings of

the 3rd Workshop on Hot Topics in Storage and File Systems (HotStorage 11). Berkeley, CA, USA: USENIX Association,2011:1-5.

[86] XU L H,BRUCK J. X-Code:MDS array codes with optimal encoding [J]. IEEE Transactions on Information Theory,1999,45(1):272-276.

[87] HAFNER J L. HoVer erasure codes for disk arrays [C]//Proceedings of International Conference on Dependable Systems and Networks (DSN 06). Washington, DC, USA: IEEE Computer Society, 2006: 217-226.

[88] HOU H X,LEE P P C. STAR+ codes:triple-fault-tolerant codes with asymptotically optimal updates and efficient encoding/decoding [C]//Proceedings of the 2021 IEEE Information Theory Workshop (ITW 21). Kanazawa,Japan:IEEE,2021:1-6.

[89] SHUM K W. Cooperative regenerating codes for distributed storage systems [C]//Proceedings of 2011 IEEE International Conference on Communications (ICC 11). New York:IEEE Communications Society, 2011:1-5.

[90] SHUM K W,HU Y C. Cooperative regenerating codes [J]. IEEE Transactions on Information Theory,2013, 59(11):7229-7258.

[91] HOU H X,LEE P P C,SHUM K W,et al. Rack-aware regenerating codes for data centers [J]. IEEE Transactions on Information Theory,2019,65(8):4730-4745.

[92] HOU H X,LEE P P C. Generalized rack-aware regenerating codes for jointly optimal node and rack repairs [C]//Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT 21). Melbourne,Australia:IEEE,2021:2191-2196.

[93] 杨松霖,张广艳. 纠删码存储系统中数据修复方法综述[J]. 计算机科学与探索,2017,11(10):1531-1544.

[94] RESCH J K,PLANK J S. AONT-RS:blending security and performance in dispersed storage systems [C]//Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST 11). Berkeley,CA,USA: USENIX Association,2011:191-202.

[95] LI M,QIN C,LEE P P C,et al. Convergent dispersal: toward storage-efficient security in a cloud-of-clouds [C]//Proceedings of the 6th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 14). Berkeley, CA, USA: USENIX Association, 2014: 1-5.

[96] LI M Q,QIN C,LI J W. CDStore:toward reliable,secure,and cost-efficient cloud storage via convergent dispersal [J]. IEEE Internet Computing,2016,20(3): 45-53.

[97] SHEN J J,GU J Z,ZHOU Y F,et al. Cloud-of-clouds storage made efficient:a pipeline-based approach [C]//Proceedings of 2016 IEEE International Conference on Web Services (ICWS 16). Washington,DC,USA:IEEE Computer Society,2016:724-727.

[98] 朱良杰,沈佳杰,周扬帆,等. 云际存储系统性能优化研究现状与展望[J]. 计算机工程与科学,2021,43(5): 761-772.

[99] SHEN L,FENG S F,SUN J J,et al. CloudS:a multi-cloud storage system with multi-level security [J]. IEICE Transactions on Information and Systems,2016, 99(8):2036-2043.

[100] ABU-LIBDEH H,PRINCEHOUSE L,WEATHERSPOON H. RACS:a case for cloud storage diversity [C]//Proceedings of the 1st ACM Symposium on Cloud Computing. New York:ACM,2010:229-240.

[101] LUO J Q,XU L H,PLANK J S. An efficient XOR-Scheduling algorithm for erasure codes encoding [C]//Proceedings of 2009 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN 09). Lisbon,Portugal:IEEE,2009:504-513.

[102] LUO J Q,SHRESTHA M,XU L H,et al. Efficient encoding schedules for XOR-based erasure codes [J]. IEEE Transactions on Computers,2013,63(9):2259-2272.

[103] HUANG C,LI J,CHEN M H. On optimizing XOR-based codes for fault-tolerant storage applications [C]//Proceedings of Information Theory Workshop (ITW 07). Tahoe City,CA,USA:IEEE,2007:218-223.

[104] HAFNER J L,DEENADHAYALAN V,RAO K K, et al. Matrix methods for lost data reconstruction in erasure codes [C]//Proceedings of the 4th USENIX Conference on File and Storage Technologies (FAST 05). Berkeley,CA,USA:USENIX Association,2005: 15-30.

[105] PLANK J S,SCHUMAN C D,ROBISON B D. Heuristics for optimizing matrix-based erasure codes for fault-tolerant storage systems [C]//Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 12). Boston,MA,USA: IEEE,2012:1-12.

[106] XIANG L P,XU Y L,LUI J C S,et al. Optimal recovery of single disk failure in RDP code storage systems [C]//Proceedings of ACM SIGMETRICS Per-

formance Evaluation Review. New York: ACM, 2010: 119-130.

[107] KHAN O, BURNS R, PLANK J, et al. Rethinking erasure codes for cloud file systems: minimizing I/O for recovery and degraded reads [C]//Proceedings of the 10th USENIX Conference on File and Storage Technologies (FAST 12). Berkeley, CA, USA: USENIX Association, 2012: 251-264.

[108] 傅颖勋,文士林,马礼,等.纠删码存储系统单磁盘错误重构优化方法综述[J].计算机研究与发展,2018,55(1):1-13.

[109] XU S L, LI R H, LEE P P C, et al. Single disk failure recovery for X-code -based parallel storage systems [J]. IEEE Transactions on Computers, 2014, 63(4): 995-1007.

[110] LI S Y, CAO Q, HUANG J Z, et al, PDRS: a new recovery scheme application for vertical RAID-6 code [C]//Proceedings of 2011 IEEE Sixth International Conference on Networking, Architecture, and Storage (NAS 11). Washington, DC, USA: IEEE Computer Society, 2011: 112-121.

[111] ZHANG G Y, WU G Y, WANG S P, et al. CaCo: an efficient cauchy coding approach for cloud storage systems [J]. IEEE Transactions on Computers, 2016, 65(2): 435-447.

[112] LI J, SHUANG Y, WANG X, et al. Tree-structured data regeneration with network coding in distributed storage systems [C]//Proceedings of the 17th International Workshop on Quality of Service (IWQoS 09). Charleston, SC, USA: IEEE, 2009: 1-9.

[113] PEI X Q, WANG Y J, MA X K, et al. Cooperative repair based on tree structure for multiple failures in distributed storage systems with regenerating codes [C]//Proceedings of the 12th ACM International Conference on Computing Frontiers. New York: ACM, 2015: 1-8.

[114] WANG Y, WANG X. A fast repair code based on regular graphs for distributed storage systems [C]// IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum. Washington, DC, USA: IEEE Computer Society, 2012: 2486-2489.

[115] SHAH N B, RASHMI K V, VIJAY KUMAR P. A flexible class of regenerating codes for distributed storage [J]. IEEE International Symposium on Information Theory (ISIT 10). Austin, Texas, USA: IEEE, 2010: 1943-1947.

[116] SHEN J J, GU J Z, ZHOU Y F, et al. Flexible regenerating codes for multiple node failures [C]//Proceedings of CCF Conference on BigData (CCF BigData 16). Lanzhou, Gansu, China: CCF, 2016: 1-10.

[117] SHEN Z R, LEE P P C, SHU J W. Efficient routing for cooperative data regeneration in heterogeneous storage networks [C]//Proceedings of IEEE/ACM 24th International Symposium on Quality of Service (IWQoS 16). Beijing, China: IEEE, 2016: 1-10.

[118] PLANK J S, THOMASON M G. A practical analysis of low-density parity-check erasure codes for wide-area storage applications [C]//Proceedings of the 2004 International Conference on Dependable Systems and Networks (DSN 04). Washington, DC, USA: IEEE Computer Society, 2004: 115-124.

[119] MITRA S, PANTA R, RA M-R, et al. Partial-Parallel-Repair (PPR): a distributed technique for repairing erasure coded storage [C]//Proceedings of the Eleventh European Conference on Computer Systems (EuroSys 16). New York: ACM, 2016: 1-16.

[120] RASHMI K V, NAKKIRAN P, WANG J, et al. Having your cake and eating it too: jointly optimal erasure codes for I/O, storage, and network-bandwidth [C]// Proceedings of the 13th USENIX Conference on File and Storage Technologies (FAST 15). Berkeley, CA, USA: USENIX Association, 2015: 81-94.

[121] RASHMI K V, SHAH N B, VIJAY KUMA P. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction [J]. IEEE Transactions on Information Theory, 2011, 57(8): 5227-5239.

[122] YAO Q R, HU Y C, TU X Y, et al. PivotRepair: fast pipelined repair for erasure-coded hot storage [C]// 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS). Bologna, Italy: IEEE, 2022: 1-11.

[123] 钟凤艳,王艳,李念爽.异构环境下纠删码的数据修复方法综述[J].计算机应用研究,2019,36(8):2241-2249.

Research Progress and Prospect on Performance Optimization of Data Recovery for Erasure-Coded Storage Systems

SHEN Jiajie,XIANG Wang**,SHEN Minhu,WU Bochun,ZHAO Zeyu,ZHANG Kai
(Informatization Office,Fudan University,Shanghai,200433,China)

Abstract:Erasure codes are widely used in distributed storage systems to store user data for online applications. When some storage nodes fail,the erasure-coded storage systems need to replace the original failed nodes with new storage nodes and recover the failed user data. Since the erasure-coded storage systems need to perform the data encoding,transmission and read-write operations,erasure code storage systems usually take a long time to perform data repair operations,and the stored user data will be unreliable for a long time. In order to ensure the reliability of stored data,researchers have proposed a variety of data recovery performance optimization schemes to reduce data repair time. The problem of data recovery performance optimization were introduced,the main performance bottlenecks and performance optimization difficulties in various application scenarios were analyzed,the main technical implementation schemes and research work to improve the data recovery performance were summarized,the future development direction of the data recovery performance optimization research field was prospected,which provided ideas for the designers of erasure-coded storage system to accurately select the data recovery performance optimization scheme suitable for specific application scenarios.

Key words:distributed storage systems;erasure-coded storage systems;data recovery operation;data reliability;performance optimization

责任编辑:唐淑芬