背景综述

# 一、背景简介

随着计算机技术和网络技术地发展，数据正以爆炸式的速度增长，海量信息对存储系统提出了巨大的挑战。特别是随着存储系统总存储介质数目的增加，以及存储介质的多样化和复杂化，存储系统的存储介质错误或者存储介质中潜在扇区错误出现的概率也越来越高。为了提高数据的可靠性和可用性，需要研究高效并易于实现的存储容错技术。目前，这些技术都是通过增加冗余的信息来查找并修复存储介质故障的。

通常的冗余备份机制包括两类：一类是完全的数据备份机制，即镜像方法；而另一类就是纠删码的方法。

镜像方法又称为多副本技术，就是把数据复制多个副本分别存储起来，以实现冗余备份。这种方法不涉及专门的编码和重构算法，容错性能比较好，但存储利用率极低，当存放N个副本时，磁盘利用率仅有1/N，尤其是当系统规模很大时，镜像技术带来的额外存储空间的开销很大，导致存储成本非常高。

纠删码起源于通信传输领域，起初主要用于解决数据传输中的检错和纠错问题。后来纠删码逐渐应用到存储系统中的数据检错和纠错问题中，以提高存储系统的可靠性，并根据存储系统应用的特点逐步得到改进和推广。在存储系统中，纠删码技术主要是通过利用纠删码算法将原始的数据进行编码得到冗余，并将数据和冗余一起存储起来，以达到容错的目的。其基本思想是将k块原始的数据元素通过一定的编码计算，得到m块冗余元素。对于这k+m块的元素，当其中任意的m块元素出错（包括数据和冗余出错）时，均可以通过对应的重构算法恢复出原来的k块数据。机遇纠删码的方法与传统的镜像副本技术相比，具有冗余 度低，磁盘利用率高等优点。

# 二、典型纠删码

1. RS（Reed-Solomon）编码

RS编码是唯一可以满足任意数据磁盘数目（n）和冗余磁盘数目（m）的达到理论上最优存储利用率的编码。RSb编码起源于1960年，经过长期的发展，已经具有较为完善的理论基础。RS编码是在Galois域GF（2w）上进行所对应的域元素的多项式运算（包括加法运算和乘法运算）的编码方式，它属于横式编码，即冗余独立于数据存储。RS编码通常分为两类：一类是范德蒙RS编码，另一类是柯西RS编码。范德蒙RS编码使用的生成矩阵是范德蒙矩阵，而柯西编码所用的生成矩阵是柯西矩阵。

1. 横式阵列码

与RS编码相比，阵列码完全基于异或（XOR）运算。阵列码的含义就是将原始的数据和冗余都存储在一个2维（或者多维）的阵列中。与传统的RS编码相比，在阵列码中，仅使用异或操作，实现容易，编码，更新以及重构的过程相对比较简单。横式阵列码是指冗余独立于数据条块，单独存储在冗余条块的阵列编码方式，他的结构特点是让冗余单独存放在独立的磁盘中（这些磁盘被称为冗余磁盘），而剩下的磁盘专门用于存储数据，这样的排布方式使得整个磁盘阵列具有非常好的可扩展性。比如EVENODD，RDP，EEO等都属于横式阵列码。

1. 纵式阵列码

纵式阵列码是指冗余存储在数据条块中的阵列编码方式，即在纵式编码中的某些条块中既存储了数据元素，又存储了冗余元素。纵式阵列码一般都具有较简单地几何结构，其计算复杂度的开销均匀地分摊到各个磁盘上。比如X-code，P-code，WAVER等都属于纵式阵列码。

1. LDPC编码

LDPC（low-density parity-check code）码是一类所有运算过程完全基于异或（XOR）运算的编码。LDPC编码的编码方式实际上是构建在一个二部图之上。比较典型的LDPC编码是构建在Tanner之上的编码方式，在这种编码方式中，二部图的左端包括了所有的数据元素和冗余元素，右端的每个点则代表了一个等式关系，表示与其相连的左端的各数据元素的异或之和为0。LDPC编码最大的特点就是过程简单，速度很快。

# 三、主要挑战

采用纠删码进行数据容错不仅可以极大地降低系统的存储空间消耗 ， 甚至同时还可以提供更高的数据可靠性。早期纠删码主要应用在基于RAID的存储系统中。由于以前芯片的运算能力有限，早期关于纠删码的研究主要关注其计算效率问题。为此，涌现出了众多只需进行异或运算的纠删码，极大降低了计算的复杂度。

当前，纠删码容错技术在大规模分布式存储中面临的主要挑战已不再是其较高的运算复杂度，而是其较高的网络资源消耗，以及实现上的复杂性。一方面，随着CPU运算能力的飞速发展，现在普通商用服务器的运算能力对于大部分存储系统已经严重过剩。例如，百度云盘的云端存储系统ATLAS甚至由于普通处理器利用率过低而采用了ARM处理器。另一方面，与 RAID的集中式操作不同，分布式系统的分布特性使得纠删码容错的相关操作需要多个节点相互协作，不可避免地带来大量的数据传输，占用较多的网络资源。一直以来，网络都是分布式系统中的稀缺资源，往往是整个系统性能的瓶颈所在。此外，分布式系统的特性也增加了实现的难。具体来说，大规模分布式存储系统中纠删码容错技术面临的挑战主要表现在编码实现，数据修复和数据更新等3个方面。