Erasure Codes for Storage Systems Summary

本Summary以Erasure Codes for Storage Systems Summary的内容为主，补充了部分关于$GF(2^w)$域上的计算、RAID系统的分类等相关内容。关于纠删码的相关知识概述也可参考存储系统中的纠删码-综述

简单纠删码示例

纠删码：与纠错码、检错码类似，均为线性分组码，通过编码可以在有限损失的前提下恢复丢失的数据。

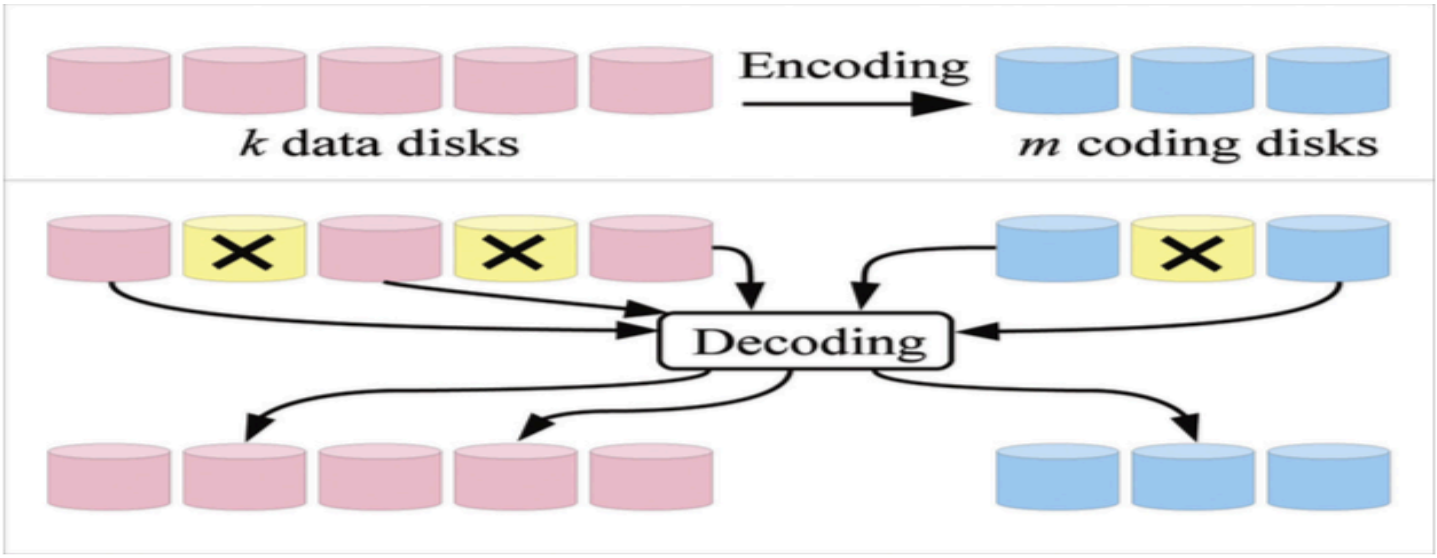
相关参数

以磁盘做为单位存储设备的存储系统中，有以下参数说明：

- n : 存储系统中磁盘总数
- k : 原始数据盘个数或恢复数据需要的磁盘
- m : 校验盘个数$m = n - k$

编码策略编码k个数据盘，得到m个校验盘，保证丢失若干个磁盘可以恢复出丢失磁盘数据，编码效率$R = \frac{k}{m}$（磁盘可以推广为数据块或者任意存储节点）。

基本操作



假设每个磁盘存储w比特数据，设d_0，$\cdots$，d_{k-1}是存储在k个数据磁盘上的数据，

c_0，$\cdots$，c_{m-1}是存储在m个编码盘上的编码。编码定义为数据的线性组合：

$$
\begin{aligned}
c_0 &= a(0,0)d_0 + \cdots + a(0,k-1)d_{k-1} \\
c_1 &= a(1,0)d_0 + \cdots + a(1,k-1)d_{k-1} \\
&\cdots\cdots \\
c_{m-1} &= a(m-1,0)d_0 + \cdots + a(m-1,k-1)d_{k-1}
\end{aligned}
$$

编码只需要乘法和加法（$w=1$时加法指模2加 / 异或、乘法指二进制与运算），解码需要用高斯消除或矩阵求逆求的方法解一组线性方程。

> 译码准则：最小差错概率译码、最大似然译码

以上计算建立在$GF(2^w)$上，这里的w值一般选取2的幂，较为流行的值有$w=1$（计算简单）$w=8$（一个字节）w值越大，纠删码越丰富，但随w值增大，在伽罗瓦域上的计算复杂度不断提高。

$GF(2^w)$扩展域上的计算

加法、减法

假设$A(x), B(x) \in GF(2^w)$，求和结果为$C(x)$。

$$C(x) = A(x) + B(x) = \sum_{i=0}^{w-1} c_i x^i, \quad c_i \equiv a_i + b_i mod 2$$

乘法

假设$A(x), B(x) \in GF(2^w)$，且

$$P(x) = \sum_{i=0}^{w-1} p_i x^i, \quad p_i \in GF(2)$$

是一个不可约多项式。两个元素的乘法运算定义为：

$$C(x) = A(x) \cdot B(x) mod P(x)$$

求逆

在乘法中定义的$P(x)$基础上，任一非零元$A \in GF(2^w)$的逆元定义为：

$$A^{-1}(x) \cdot A(x) \equiv 1 mod P(x)$$

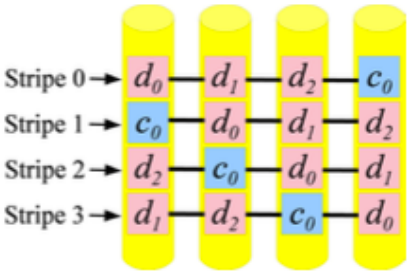
最大距离可分码（MDS）

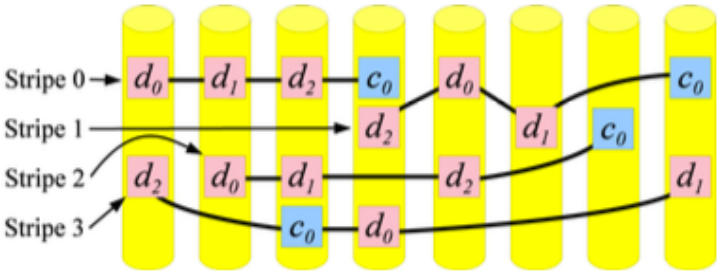
一个$(n, n - r, r + 1)$码称为最大距离可分（Maximum Distance Separable, MDS）码。一个MDS码是冗余度为r，最小距离等于r＋1的线性码。（实现最佳容错能力）

数据块大小和条带（stripe）

数据块大小是数据块进行编码计算时，数据分块的大小。在不同的存储系统中数据块大小不同，在RAID存储系统中，常见的分块大小是4KB~128KB，在分布式存储系统中数据块较大，一般为64MB大小。

条带是纠错编码存储一次性读入或写入数据的大小，在系统码中，一个条带往往包含了k个原始数据块和m个校验块。





上图左侧，其中$n = 4$个磁盘中的每个磁盘包含相同比例的数据和编码条带。
上图右侧，包含$n = 8$个磁盘，但每个条带由三条数据条带和一条编码条带组成，条带的分配是唯一的区别。因此，擦除码可以与图的左侧相同。（Panasas使用这种方法来允许灵活的块分配，并允许将额外的磁盘无缝地添加到存储系统）

RAID

更多内容可以参考RAID技术规范中相关的内容，讲解较为详细。

RAID技术

RAID简介

冗余磁盘阵列技术最初的研制目的是为了组合小的廉价磁盘来代替大的昂贵磁盘，以降低大批量数据存储的费用，同时也希望采用冗余信息的方式，使得磁盘失效时不会使对数据的访问受损失，从而开发出一定水平的数据保护技术，并且能适当的提升数据传输速度。

RAID规范

主要包含$RAID0 \sim RAID7$等数个规范，它们的侧重点各不相同，常见的规范有如下几种：

RAID 0

实现需要至少两块磁盘，没有差错控制、数据分布于不同磁盘上，数据吞吐能力强。设磁盘数为n，则读些速度均提高为单个磁盘的n倍。

RAID 1

采用镜象结构，用n个磁盘形成n份完整备份，安全性高，读写速度相对单个磁盘没有优化，在某个磁盘故障后可从其他磁盘恢复，同时应当及时更新损坏的磁盘。

RAID 2, 3

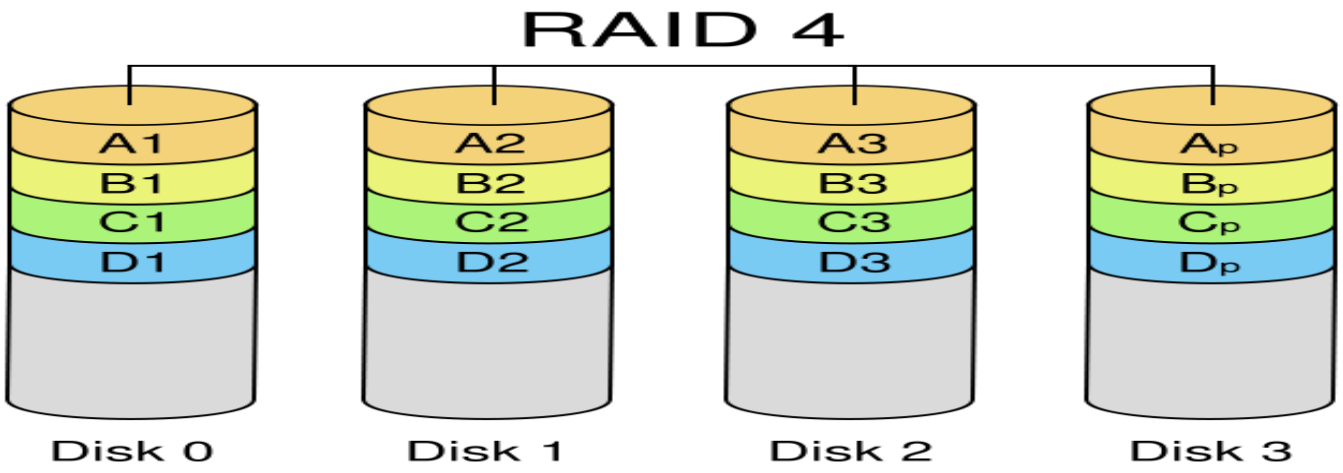
RAID 2使用汉明码校验，具有检错、纠错能力；
RAID 3使用奇偶校验码，具有检错能力但不可纠错。

RAID 4, 5

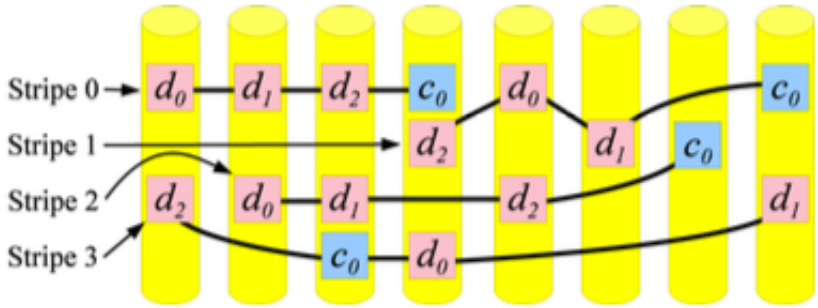
RAID 4和RAID 5使用相同的简单擦除代码，但具有不同的条带布局。在这里，纠删码是一种MDS码，其中$m = 1$，$w = 1$，唯一的编码位被标记为p，它是所有数据位的异或：

$$p = d0 \oplus d1 \oplus \ldots \oplus dk - 1$$

当任何位被擦除时，它可能被解码为幸存的位异或。
RAID 4构成方式如下所示，每个磁盘身份固定，有一个磁盘p专用于编码。



RAID 5构成方式如下所示，磁盘的身份在各个条带中轮转，每个磁盘都保存有数据和编码，系统更加平衡。



RAID 6

添加一个磁盘(Q)到RAID-4/5系统中，该系统使用6个磁盘进行存储。使用一个(6,2)MDS码。以$w = 8$为例，通过以下算法进行计算：

$$p = d_0 \oplus d_1 \oplus \cdots \oplus d_{k-1}$$

$$q = d_0 \oplus 2(d_1) \oplus \cdots \oplus 2^{k-1}(d_{k-1})$$

其中，由于伽罗瓦域中加法相当于异或操作，因此P盘的纠错码相当于RAID-4/5，同时由于：

$$q = 2(2(\cdots 2(2d_{k-1} \oplus d_{k-2}\ ) \cdots) \oplus d_1\ ) \oplus d_0$$

所以Q盘仅可以使用加法和乘法进行计算。

编码示例

Reed-Solomon Codes（RS码）

RS码是当且仅当$n \leqslant 2^w$时存在的MDS码。例如，存储系统包含256个或更少的磁盘，可以在$GF(2^8)$中定义一种计算。有多种方式来定义$a(i,j)$系数。

最简单是“Cauchy”结构：在$GF(2^w)$中选择n个不同的数字，并将它们分成两组X和Y，使得X具有m个元素，Y具有k个元素。计算纠删码系数：

$$a(i,j) = \frac{1}{x_i \oplus y_j}$$

Array Codes（阵列码）

- 避免伽罗瓦域计算
- 仅通过异或运算实现纠错码

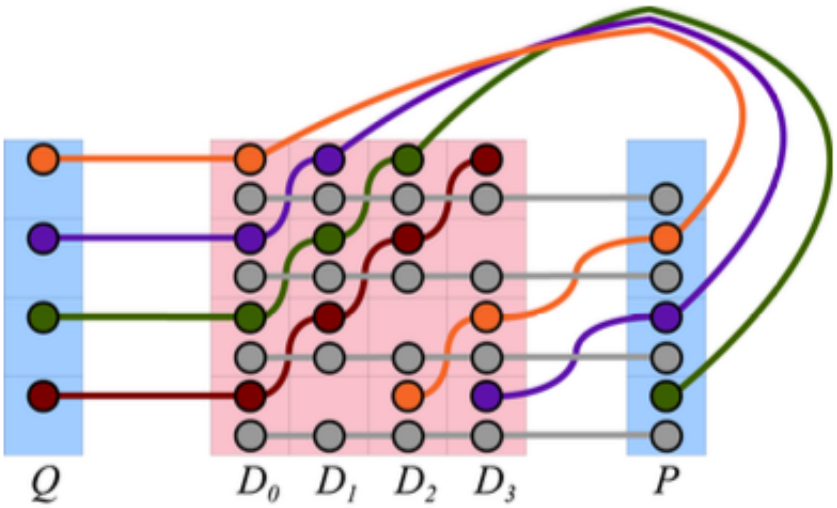
编码字节是数据字节的线性组合。

知名方案:

- 用于RAID-6的RDP，EVENODD，Blaum-Roth和Liberation-codes。
- 用于$m = 3$的STAR阵列码；
- 用于任意k和m值的Cauchy Reed-Solomon，通用EVENODD、通用RDP。

RAID 6示例

使用RDP阵列码的实现中，$k = 4, m = 2, n = k + m = 6$, $r = 4$, $w = 1$。如下图所示，灰线描绘了P盘的编码方程。其他的线描述了Q盘的编码方程。

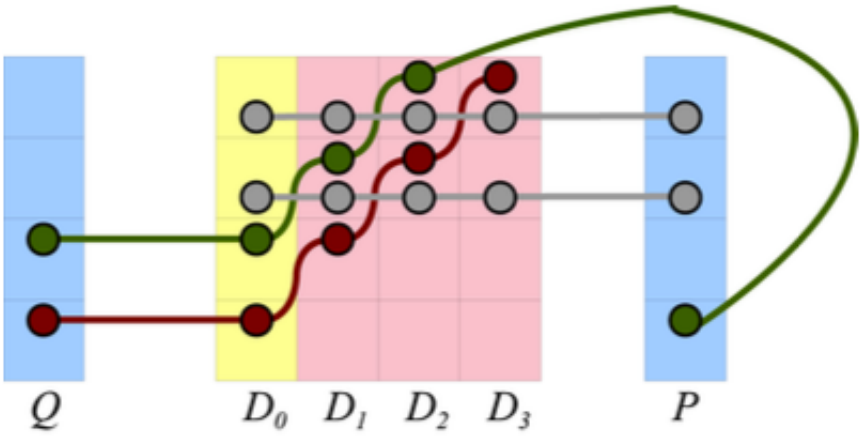


对存储系统的改进成果

恢复时减少磁盘I/O消耗

当使用简单纠删码且单个磁盘发生故障时，恢复其内容需要从幸存的磁盘读取$k - 1$个条带，以计算故障磁盘上的每个条带。

使用阵列代码，可以显着减少从幸存的磁盘读取的数据量。



如上图所示的例子中（使用RDP纠删码），磁盘D_0损坏时，若使用简单纠删码，恢复将等效于仅从P驱动器进行解码，必须从幸存的磁盘读取16bit; 但由于RDP的结构，通过从幸存的磁盘读取12bit来恢复D_0。这种方法意味着恢复时可以减少25%的磁盘I / O操作。

再生编码

再生编码可以在使用纠删码的存储系统中减少其恢复时占用的网络I/O。当一个或多个存储节点发生故障时，系统可以使用保存其先前内容的节点或通过纠删码保存了等效内容的节点来进行替换。

非MDS纠删码

由于非MDS码不能应对m个节点故障的所有可能情况，所以需要添加额外的存储设备来满足m个磁盘损坏时数据恢复的需求。但同时可以带来显著的性能改进。是一种利用冗余存储空间换取恢复性能的方法。