

RAID6 编码的扩展算法及性能研究

金 超,冯 丹,刘景宁,田 磊

(华中科技大学计算机科学与技术学院,武汉光电国家实验室,湖北武汉 430074)

摘 要: RAID6 编码根据其码字结构可以分为水平码和垂直码两大类. RAID6 水平码可以很容易的扩展至任意码长,而 RAID6 垂直码通常具有码长的限制. 本文提出一种针对 RAID6 垂直码的码长扩展算法,该算法通过校验块变更的方法,可以将 RAID6 垂直码扩展至任意码长. 该算法可以保持 RAID6 垂直码的 MDS 特性. 本文研究了 RAID6 编码的扩展算法在应用于各种 RAID6 编码时对其性能的影响,揭示出 RAID6 编码在进行扩展时的性能变化规律.

关键词: 独立冗余磁盘阵列级别 6 (RAID6) 编码; 水平码; 垂直码; 码长扩展; 性能分析

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2012) 01-0173-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.01.028

Code Shortening and Performance Analysis for RAID6 Codes

JIN Chao, FENG Dan, LIU Jing-ning, TIAN Lei

(School of Computer Science and Technology, Huazhong University of Science and Technology,
Wuhan National Laboratory for Optoelectronics, Wuhan, Hubei 430074, China)

Abstract: There are generally two categories of RAID6 codes in their code structures, horizontal codes and vertical codes. Horizontal codes are very easy to be extended to an arbitrary length, while the lengths of vertical codes are usually restricted. We propose a novel and efficient code shortening algorithm for vertical codes, by which the length of a vertical code can be extended to an arbitrary given one. The shortening algorithm maintains the MDS (Maximum Distance Separable) property of the vertical codes. In the context of our shortening algorithm for vertical codes, we conduct an performance analysis on the representative vertical and horizontal RAID6 codes at all code lengths, and reveal the impacts of the code shortening algorithms on the performance of the RAID6 codes in various performance metrics.

Key words: redundant array of independent disk 6 (RAID6) codes; horizontal codes; vertical codes; code shortening; performance analysis

1 引言

磁盘阵列中的 RAID6 结构在阵列中任意双盘同时失效后仍能恢复数据,因而能够显著提高存储系统的可靠性. RAID6 阵列的容错能力来自于底层的 RAID6 编码^[1]. RAID6 编码的性能在很大程度上决定了 RAID6 阵列的整体性能. RAID6 编码的性能通常有三个衡量标准:编解码计算复杂度,更新复杂度,存储效率^[2]. 编解码计算复杂度正比于在编解码过程中所需异或计算的次数. 更新复杂度是指每更新一个数据块所需要同时更新的校验块的个数. 存储效率是指 RAID6 编码结构中数据块所占的比例. 对于包括 RAID6 编码在内的所有纠删码,其容错能力和存储效率之间的关系都满足 Singleton 不等式^[3]. 特别的,达到 Singleton 界的纠删码被称

为 MDS 编码, MDS 编码具有最优的存储效率.

RAID6 编码可分为水平码和垂直码两大类. 在水平码结构中,数据块和校验块分别放置在数据列和校验列中. 在垂直码结构中没有单独的校验列,校验块被分散放置于各个数据列中. RAID6 编码在原始定义时通常都具有码长的限制. 比如,标准 RDP (Row Diagonal Parity) 码^[4]的码长必须为素数加 1, 标准 X-Code 码^[5]的码长必须为素数. 水平码可以很容易的通过削减数据列的方式来扩展至任意码长,而垂直码却不能简单的通过这种方式来进行扩展,因为垂直码的数据列中含有校验块. RAID6 编码的码长对应于 RAID6 阵列中磁盘的个数,因此,如果不对 RAID6 垂直码的码长进行扩展,将会极大的限制其在 RAID6 阵列中的应用.

本文提出了一种针对 RAID6 垂直码的码长扩展算

法,该算法通过变更校验块的方法,能够将 RAID6 垂直码扩展至任意码长.并且,扩展后的 RAID6 垂直码与标准 RAID6 垂直码一样,是 MDS 编码. RAID6 编码在进行码长扩展时其性能会发生变化.本文分别研究了通用的 RAID6 水平码扩展算法和本文提出的 RAID6 垂直码扩展算法在应用于各种 RAID6 编码时对其性能的影响,揭示了 RAID6 编码在码长扩展时的性能变化规律.最后,本文介绍了 RAID6 垂直码的“垂直削减”算法,并揭示了 RAID6 垂直码在进行垂直削减时其存储效率、编解码计算复杂度和数据恢复效率的变化规律.

2 RAID6 编码

本节通过一种基于标签的方式来分析水平码(RDP 码和 Liberation 码)及垂直码(P-Code 码)的结构.编码结构中每个数据或校验块对应一个标签,每个标签由若干个整数组成,其中每个整数代表该数据或校验块所参与的一个校验条带的序号.在本文下面的叙述中, p 代表某个素数.

RDP 码^[4].标准的 RDP 码含有 $(p-1)$ 行和 $(p+1)$ 列,其编码结构如图 1 所示. RDP 结构中存在一条“缺失”对角线,该对角线上的数据块仅参与一条行校验条带,而不参与任何对角线校验条带.如列 d_4 中的数据块(2)仅参与行校验条带 $P(2)$.另外, RDP 码中的每个行校验块(除第一个外)同时参与一条对角线校验条带. RDP 码达到了最低的编解码计算复杂度,但其更新复杂度未达到最低.

d_1	d_2	d_3	d_4	P	Q
1,5	1,6	1,7	1,8	1	5
2,6	2,7	2,8	2	2,5	6
3,7	3,8	3	3,5	3,6	7
4,8	4	4,5	4,6	4,7	8

图1 含有6列的标准RDP编码结构

Liberation 码^[6].标准的 Liberation 码含有 p 行和 $(p+2)$ 列. Liberation 码的结构如图 2 所示.事实上,每个 RAID6 水平码都可以表示成生成矩阵和数据向量乘积的形式. Liberation 码的生成矩阵中含有 1 的个数在所有 RAID6 水平码中是最少的,意味着 Liberation 码在所有 RAID6 水平码中具有最低的更新复杂度.但 Liberation

d_1	d_2	d_3	d_4	d_5	P	Q
1,6	1,10	1,9,10	1,8	1,7	1	6
2,7	2,6	2,10	2,9	2,8,9	2	7
3,8	3,7,8	3,6	3,10	3,9	3	8
4,9	4,8	4,7	4,6,7	4,10	4	9
5,10	5,9	5,8	5,7	5,6	5	10

图2 含有7列的标准Liberation编码结构

码的编解码计算复杂度却不是最低的.

P-Code 码^[7].标准的 P-Code 码含有 $(p-1)/2$ 行和 $(p-1)$ 列.如图 3 所示, P-Code 的校验块放在编码结构的第一行,其余行存放数据块.每个数据块的标签由两个整数组成,并且这两个整数模 p 的和与同列中校验块标签所含的整数相等.标准的 P-Code 达到了最低编解码计算复杂度和最低更新复杂度.

d_1	d_2	d_3	d_4	d_5	d_6
1	2	3	4	5	6
2,6	3,6	1,2	1,3	1,4	1,5
3,5	4,5	4,6	5,6	2,3	2,4

图3 含有6列的标准P-Code编码结构

3 RAID6 垂直码的码长扩展算法

RAID6 水平码很容易通过削减数据列的方法扩展至任意码长.对于一个标准的 RAID6 水平码编码结构,假设其中某些数据列固定存放全零数据块,那么这些数据列可以直接从编码结构中移除,因为全零数据块对校验条带的异或计算不产生影响.这样得到的编码结构便是码长比标准 RAID6 水平码要短的扩展 RAID6 水平码.要生成指定码长的 RAID6 水平码,总可以找到一个码长更长的标准 RAID6 水平码,再通过削减数据列的方式得到该码长的水平码.

在 RAID6 垂直码的编码结构中,每个数据列既包含数据块又包含校验块.如果直接移除某个数据列,会使某些校验条带失去校验块.因此, RAID6 垂直码不能简单的通过削减数据列的方法来进行扩展.本节将介绍一种适合于 RAID6 垂直码的码长扩展算法,该算法通过变更校验块的方法,能够将标准 RAID6 垂直码扩展至任意码长,并且能同时保持其 MDS 特性.下面将以 P-Code 为例来描述该码长扩展算法.值得注意的是,该算法同样适用于 X-Code^[5]等其他 RAID6 垂直码.

3.1 算法描述

首先通过一个实例来介绍该算法.假设要在如图 3 所示的含有 6 列的标准 P-Code 结构的基础上扩展生成含有 5 列的 P-Code 结构,需要进行如下两步操作.首先,削去数据列 d_6 ;然后,选择数据列 d_1 中的数据块(2,6)作为校验条带 $P(6)$ 的新校验块.生成的具有 5 列的扩展 P-Code 结构如图 4 所示.与水平码的扩展算法相比, P-Code 的扩展算法增加了一个变更校验块的步

d_1	d_2	d_3	d_4	d_5
1	2	3	4	5
2,6	3,6	1,2	1,3	1,4
3,5	4,5	4,6	5,6	2,3

图4 含有5列的扩展P-Code编码结构

骤,为削去校验块的校验条带分配新的校验块.这样做保证了扩展后的 P-Code 结构中每条校验条带仍然是完整的,并且仍然具备容双列失效的能力.

算法 1 给出了构造任意给定码长 n 的 P-Code 的算法.算法首先构造一个码长为 $(p-1)$ 的标准 P-Code 结构(p 为大于 n 的素数),然后再在此基础上进行扩展得到码长为 n 的 P-Code 结构.算法的最后两步是为削去校验块的校验条带在第 1 列或第 2 列中分配新的校验块.此处需要遵循的原则是,为每个校验条带分配的新校验块必须事先是该校验条带上的一个数据块.图 5 给出了校验块的分配顺序.从 P-Code 的编码结构可以看出,第 1 列中数据块标签的第 2 个数字,以及第 2 列数据块标签的第 1 个数字,按照分配顺序刚好构成了从 $(p-1)$ 到 3 的倒序排列.因此,第 1 列和第 2 列总能在遵循分配原则的前提下为每一个削去校验块的校验条带分配新的校验块.

算法 1 构造码长为 n 的 P-Code 算法

- 1) 选择一个大于 n 的素数 p ,构造码长为 $(p-1)$ 的标准 P-Code.
- 2) 从构造的标准 P-Code 结构中按从后向前的顺序削去后面的 $(p-1-n)$ 列.
- 3) 根据分配顺序,在第 1 列或第 2 列中选择一个数据块作为校验条带 $P(p-1)$ 的新校验块.
- 4) 分别对校验条带 $P(p-2), \dots, P(n+1)$ 重复第 3 步的操作.

d_1	d_2	d_3	d_4	d_5	d_6
1	2	3	4	5	6
2,6	3,6	1,2	1,3	1,4	1,5
3,5	4,5	4,6	5,6	2,3	2,4

图 5 P-Code 扩展算法中新校验块的分配顺序(图中箭头所示)

算法 1 的第一步,素数 p 的选择具有很大的灵活性,只需保证 p 大于 n 即可.但是,参数 p 的选择会影响算法所生成的 P-Code 的性能.在对 P-Code 进行列削减时,P-Code 的性能会发生变化.选择不同的参数 p 所对应需要削减的列的数目不同,因此最后构造出的码长为 n 的 P-Code 的性能也是不同的.本文将在第 4 节中详细研究 P-Code 及其他 RAID6 编码在进行列削减时各项性能的变化规律.

3.2 算法的正确性

扩展 P-Code 同标准 P-Code 一样具有容双盘出错的能力.在标准 P-Code 结构中,任意两列的组合都对应一条“恢复链”,当这两列发生失效时,P-Code 能够按照“恢复链”的顺序逐步恢复这两列中所有数据块和校验块^[7].扩展 P-Code 恢复双列失效的方式跟标准 P-Code 完全相同.P-Code 在进行列削减时,保留列中所有块的标签都没有改变,因此,扩展 P-Code 结构中任意两列的

组合与标准 P-Code 结构中对两列的组合具有相同的“恢复链”.扩展 P-Code 与标准 P-Code 的唯一不同之处是,标准 P-Code 结构中的各个校验块是相互独立的,而扩展 P-Code 中校验块则不是独立的.比如,在图 4 中,数据块(2,6)被分配为校验条带 $P(6)$ 的新校验块,但同时它仍然参与校验条带 $P(2)$ 的计算.当 $P(6)$ 中的数据块被更新时, $P(6)$ 的校验块(2,6)需要被更新,由于(2,6)被视为 $P(2)$ 的数据块,因此 $P(2)$ 的校验块(2)也需要被更新.这样一来,与标准 P-Code 相比,扩展 P-Code 的更新复杂度和编解码计算复杂度都有所提高.但是,扩展 P-Code 与标准 P-Code 一样,仍然是 MDS 编码.本文将在第 4 节详细分析扩展 P-Code 的各项性能.

3.3 算法与另一种方案的比较

本节提出另一种扩展 RAID6 垂直码码长的算法.但是,算法 1 相比,该算法(算法 2)的缺点是其不能保持 RAID6 垂直码的 MDS 特性.

算法 1 在对 RAID6 垂直码进行扩展时通过变更校验块的方式保持了其结构中每条校验条带的完整性.而算法 2 的做法是,对于每个被削去校验块的校验条带,同时削去编码结构中所有处于该校验条带上的数据块.如图 6 所示,当削去 P-Code 编码的第 6 列时,校验条带 $P(6)$ 的校验块被削去,然后,削去所有处于 $P(6)$ 上的数据块(即标签中含数字 6 的块).为了保持每列中数据块的数目相等,额外削去第 5 列中的数据块(2,3).类似的,这些被削去的数据块可以看作是编码结构中的全零数据块,由于全零数据块对校验条带的异或计算没有影响,因此削去它们不会影响编码结构的正确性.

d_1	d_2	d_3	d_4	d_5	d_6
1	2	3	4	5	6
3,5	4,5	1,2	1,3	1,4	1,5
2,6	3,6	4,6	5,6	2,3	2,4

— 被削去的数据/校验块

图 6 另一种 RAID6 垂直码扩展算法

但是,算法 2 所生成的扩展 P-Code 不是 MDS 码.一般情况下,每削去 P-Code 结构中的一列,都要削去一个校验条带的校验块,同时剩下的每一列中都要削去一个数据块.因此,可以计算出算法 2 所生成的扩展 P-Code 的存储效率如表达式(1)所示.可以看出,这种形式的扩展 P-Code 的存储效率总是低于 RAID6 编码的最优存储效率 $(n-2)/n$ (即 MDS 码的存储效率).

$$1 - \frac{2}{n - (p - 1 - n)} \quad (1)$$

在算法 2 生成的扩展 P-Code 结构中,各校验块之间仍然是独立的,每个数据块只参与两个校验条带的

计算,因此,其更新复杂度仍然保持最优值 2. 另外需要指出的是,算法 2 保持了扩展 P-Code 结构中每列所含校验块的数目相等,而算法 1 使得第 1 列或第 2 列所含的校验块多余其他列. 在将 P-Code 应用于 RAID6 阵列时, P-Code 结构中的列和 RAID6 阵列中的磁盘之间的对应关系并不是固定的,因此,对于算法 1 所生成的 P-Code 结构,可以像 RAID5 那样,通过旋转轮换 P-Code 结构中的列和 RAID6 阵列磁盘之间的对应关系,使得 RAID6 阵列中每个磁盘所含的校验块的数目相等.

4 RAID6 编码的性能分析

本节通过对常见 RAID6 水平码(RDP 码和 Liberation 码)及垂直码(P-Code 码)进行量化分析,揭示出这些编码在进行码长扩展时各项性能的变化规律.

4.1 RAID6 编码各项性能的理论最优值

作为一类能容双列失效的纠删码, RAID6 编码同其他种类的纠删码一样,其编码结构必须满足 Singleton 不等式. 根据 Singleton 不等式可以推出,码长为 n 的 RAID6 编码成为 MDS 码的充分必要条件是,平均每 n 个块中含有 2 个校验块和 $(n-2)$ 个数据块^[7]. 因此, RAID6 编码的最高(优)存储效率为 $(n-2)/n$.

文献[7]从理论上证明了所有码长为 n 且具有 MDS 特性的 RAID6 编码的最低(优)编码复杂度是,在编码过程中平均每个数据块进行 $2-2/(n-2)$ 次异或操作,而最低(优)的解码复杂度是,在解码过程中平均每个数据块进行 $(n-3)$ 次异或操作.

另外,为了达到能容双列失效的功能, RAID6 编码结构中每个数据块必须至少参与两个校验块的计算. 因此, RAID6 编码的最低(优)更新复杂度为 2.

4.2 编解码计算复杂度分析

RDP 码在标准码长 $(p+1)$ 下具有最低编码复杂度. 同时, RDP 码在削去第 1 列后码长为 p 时仍然具有最低编码复杂度^[6]. 根据 4.1 节给出的结论,码长为 p 时 RDP 的编码复杂度为 $2-2/(p-2)$,而此时 RDP 结构中一共有 $(p-1)(p-2)$ 个数据块,因此在编码过程中所需要的异或计算总数为 $2(p-1)(p-3)$. 码长为 n 的 RDP 编码是在码长为 p 的 RDP 编码的基础上削减 $(p-n)$ 个数据列而得到的. 如图 1 所示,每个被削减的列中含有 $(p-1)$ 个数据块,这些数据块中有 1 个数据块只参与 1 条校验条带的计算,其余数据块都参与 2 条校验条带的计算. 由于削去这些数据块, RDP 在编码过程中减少的异或计算次数为 $(p-n)(2p-3)$,因此码长为 n 的扩展 RDP 码的编码复杂度可表示如下:

$$\frac{2(p-1)(p-3)-(p-n)(2p-3)}{(p-1)(n-2)} \quad (2)$$

经过类似的分析,可以得出码长为 n 的扩展 Liber-

ation 码的编码复杂度可表示如下:

$$\frac{(n-3)(2p+1)}{p(n-2)} \quad (3)$$

码长为 n 的扩展 P-Code 码的编码复杂度可表示如下:

$$\frac{(p-1)(p-4)-(p-1-n)(p-2)}{(p-1)(p-3)/2-(p-1-n)(p-1)/2} \quad (4)$$

图 8 显示了 RDP 码, Liberation 码和 P-Code 码在码长为 4 到 30 时的编码复杂度. 所有数值都是相对于同等码长下 RAID6 编码的最低编码复杂度进行归一化后的值. 每种编码对应两条曲线. 以 P-Code 为例, P-Code 所代表的曲线表示在构造码长为 n 的 P-Code 时参数 p 总是选择最接近 n 的素数,而 P-Code $p=31$ 所代表的曲线则表示参数 p 固定选择素数 31.

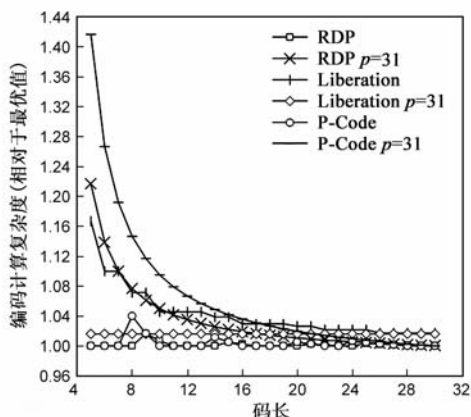


图7 RDP, Liberation Code, P-Code 的编码复杂度比较

从图 7 可以看出, RDP 和 P-Code 在各自的标准码长上达到了 RAID6 编码复杂度的最低值, 在其他码长上也非常接近于最低值. Liberation 码的编码复杂度稍高, 比最低值高出约 16%. 另一方面, 当素数 p 固定选择 31 时, RDP 和 P-Code 的编码复杂度会随着码长的减小而升高, 而 Liberation 码的编码复杂度稳定在一个很接近最低值的水平. 因此, 在进行码长扩展时, 为了使生成的 RAID6 编码的编码复杂度尽可能低, RDP 和 P-Code 需要选择尽可能小的参数 p , 而 Liberation 码则需要选择尽可能大的参数 p .

至于 RAID6 编码的解码复杂度, 由于其与编码复杂度呈现类似的规律, 在这里不再赘述.

4.3 更新复杂度分析

如图 1 所示, 在码长为 $(p+1)$ 的标准 RDP 码的编码结构中, 第 1 行以及“缺失对角线”上的每个数据块的更新复杂度为 2, 而其余数据块的更新复杂度为 3. 因此, 标准 RDP 码的平均更新复杂度可表示如下:

$$\frac{2 \times (2p-3) + 3 \times \{(p-1)^2 - (2p-3)\}}{(p-1)^2} \quad (5)$$

而对于码长为 n 的扩展 RDP 码, 其每个数据列中

含有 2 个更新复杂度为 2 的数据块和 $(p-3)$ 个更新复杂度为 3 的数据块,因此其平均更新复杂度可表示如下:

$$\frac{2 \times 2 + 3 \times (p-3)}{(p-1)} \tag{6}$$

类似的,可以分析出码长为 n 的扩展 Liberation 码的平均更新复杂度可表示如下:

$$\frac{3 \times (n-3) + 2 \times \{p(n-2) - (n-3)\}}{p(n-2)} \tag{7}$$

在标准 P-Code 码的结构中,每个数据块的更新复杂度均为 2,因此标准 P-Code 的平均更新复杂度为 2,达到了 RAID6 编码的最低更新复杂度.而扩展后的 P-Code 码的更新复杂度却不再为 2,因为其编码结构中的校验块不再是相互独立的.扩展 P-Code 码的平均更新复杂度可计算如下:

$$\begin{cases} 2 + \frac{2(p-1-n)(n-1) - (p-n)}{(p-1)(n-2)}, & n \text{ 为奇数} \\ 2 + \frac{2(p-1-n)(n-1) - (p-1-n)}{(p-1)(n-2)}, & n \text{ 为偶数} \end{cases} \tag{8}$$

图 8 显示了 RDP 码, Liberation 码和 P-Code 码在码长为 4 到 30 时的平均更新复杂度.从图中可以看出 Liberation 码的平均更新复杂度接近最低值 2,而 RDP 编码的平均更新复杂度却随着其码长的增大而上升,最后趋近于数值 3. P-Code 码的平均更新复杂度在其标准码长上达到了最低值 2,而在其他码长上比 2 稍大.另一方面,当参数 p 固定选择参数 31 时, Liberation 码的平均更新复杂度更加接近最低值 2,而 RDP 码的平均更新复杂度稳定在一个接近于 3 的数值上. P-Code 码的平均更新复杂度随着其码长的减小而呈线性上升,但不会超过其极限值 4.因此,在进行码长扩展时,为了使生成的 RAID6 编码的平均更新复杂度尽可能低, RDP 和 P-Code 需要选择尽可能小的参数 p ,而 Liberation 码则需要选择尽可能大的参数 p .这一点与编解码计算复杂度的情况类似.

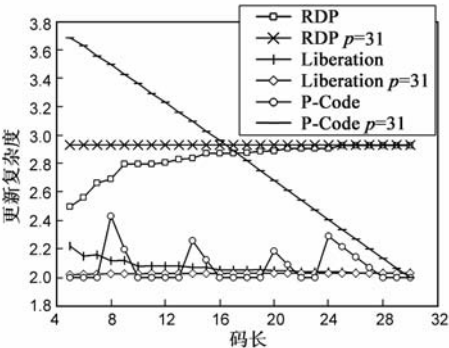


图8 RDP,Liberation Code,P-Code的平均更新复杂度比较

4.4 存储效率分析

码长为 n 的扩展 P-Code 编码结构中一共含有 $n(p-1)/2$ 个块,其中校验块的个数与标准 P-Code 编码结构相同,仍然为 $(p-1)$ 个.可以算出扩展 P-Code 编码的存储效率为 $(n-2)/n$,达到了 RAID6 编码的最优存储效率,因此扩展 P-Code 编码为 MDS 编码.

从扩展 RDP 编码和扩展 Liberation 编码的结构可以很容易的看出其存储效率仍然为 $(n-2)/n$,因此它们也都是 MDS 编码.

5 RAID6 垂直码的垂直削减算法

RAID6 编码的码长扩展算法在对编码结构进行削减时采用的是水平削减的方式,即通过移除编码结构中的某些列来减小编码结构的长度.相对于水平削减, RAID6 编码还存在垂直削减的方式,即通过移除编码结构中的某些行来减小编码结构的宽度.垂直削减的方式十分适合于 RAID6 垂直码.比如,在标准 P-Code 的编码结构中,校验块存放于第 1 行中,其他行只存放数据块.如果假定某些数据行只固定存放全零数据块,那么这些数据行可以从编码结构中移除,而不会对编码结构的正确性造成影响.另一方面,垂直削减的方式不适用于 RAID6 水平码.水平码结构的每一行都存在校验块,因此,如果移除某些行,将使某些校验条带失去校验块.

垂直削减后的垂直码不再具有最优的存储效率,因此也不再是 MDS 编码.但是,垂直削减的方式减少了垂直码在编解码过程中所需的异或计算的次数,降低了垂直码的编解码计算复杂度.因此,垂直削减方式可以通过降低垂直码的存储效率来减小垂直码的计算复杂度,为垂直码的性能提供了广阔的调节空间.

垂直削减方式的另外一个好处是,它减小了垂直码结构中的校验条带的平均长度,而更小的校验条带长度意味着在发生磁盘失效时具有更快的数据恢复速度^[8].

表 1 垂直削减前后 P-Code 的性能对比

性能指标	标准 P-Code	垂直削减后的 P-Code
平均校验条带长度	5	3
编码计算复杂度	1.5	1
更新复杂度	2	2
存储效率	67%	50%

在图 3 所示码长为 6 的标准 P-Code 结构中,如果削去最后一个数据行,则会生成含有 1 个校验行和 1 个数据行的 P-Code 结构.表 1 给出了标准 P-Code 结构和垂直削减后的 P-Code 结构各项性能指标的对比.经过垂直削减, P-Code 的存储效率降低了,但其同时也具有更低的计算复杂度和更短的校验条带长度.因此,垂直削

减后的 P-Code 在编解码时占用更少的处理器计算开销,在发生磁盘失效时也能更快的恢复数据.

6 结论

本文以 P-Code 为例提出了一种适用于 RAID6 垂直码的码长扩展算法,该算法能将 RAID6 垂直码扩展至任意码长,并能保持其 MDS 特性.同时,本文对扩展后的 RAID6 编码在计算复杂度,更新复杂度和存储效率等方面进行了深入分析,揭示出了 RAID6 编码在进行码长扩展时的性能变化规律.另外,本文还介绍了 RAID6 垂直码的垂直削减算法,显示了垂直削减可以降低 RAID6 垂直码的存储效率为代价,来换取更低的计算开销以及更快的数据恢复速度.

参考文献

- [1] 冯丹,张江陵.适用于磁盘阵列的纠双错码分析[J].电子学报,1998,26(5):86-93.
Feng Dan, Zhang Jiang-ling. Analysis of codes for tolerating double disk failures in RAID architectures[J]. Acta Electronica Sinica, 1998, 26(5): 86-93. (in Chinese)
- [2] J Plank, J Luo, et al. A performance evaluation and examination of open-source erasure coding libraries for storage[A]. Proc of FAST'09[C]. San Francisco, CA, 2009. 253-265.
- [3] M Blaum, R Roth. On lowest density MDS codes[J]. IEEE Transactions on Information Theory, 1999, 45(1): 46-59.
- [4] P Corbett. Row-Diagonal parity for double disk failure correction[A]. Proc of FAST'04[C]. San Jose, CA, 2004. 1-14.

- [5] L Xu, et al. X-Code: MDS array codes with optimal encoding[J]. IEEE Transactions on Information Theory, 1999, 45(1): 272-276.
- [6] J Plank. The RAID-6 liberation codes[A]. Proc of FAST'08[C]. San Jose, CA, 2008. 97-110.
- [7] C Jin, H Jiang, D Feng, L Tian. P-Code: A new RAID-6 code with optimal properties[A]. Proc of ICS'09[C]. New York, NY, 2009. 360-369.
- [8] M Holland, G Gibson. Parity declustering for continuous operation in redundant disk arrays[A]. Proc of ASPLOS'92[C]. Boston, MA, 1992. 23-35.

作者简介



金超男, 1983 年生于湖北随州. 博士研究生, 主要研究方向为磁盘阵列、文件系统、对象存储、云存储.

E-mail: chjinhust@gmail.com

冯丹女, 1970 年生于湖北京山. 教授, 博士生导师. 研究方向为计算机外存储系统、海量信息存储系统.

刘景宁女, 1957 年生于湖北武汉. 副教授. 研究方向为高速传输接口、FPGA 编程.

田磊男, 1978 年生于湖北武汉. 讲师. 研究方向为并行与分布式存储.