

奇校验下的三磁盘崩溃数据恢复策略^①

张 华, 冯凯平

(四川烹饪高等专科学校 信息技术系, 成都 610072)

摘 要: 灾难发生的后果可能导致数据中心存储系统中的多个数据盘同时崩溃, 如果采用镜像式备份, 极有可能导致数据无法恢复. 而采用基于奇校验的冗余技术, 在确保所有磁盘被操作概率相同的情况下, 既可以对一个或者两个崩溃的数据盘进行数据恢复, 也可以对三个或三个以上同时崩溃的数据盘进行数据恢复.

关键词: 奇校验; 数据盘; 冗余技术; 崩溃; 数据恢复

Strategy of Data Recovery for Three Collapsed Disks Under Odd Parity

ZHANG Hua, FENG Kai-Ping

(Information Technique Department, Sichuan Higher Institute Cuisine, Chengdu 610072, China)

Abstract: Some disaster may result in the simultaneous collapse of multiple data disks on the data center storage system. If a mirror backup was used, it would be highly likely that the data cannot be recovered. But if the redundancy technology, based on the odd parity, is used, and it has been ensured that all disks have equal probability of operation, then not only can the technology recover the data of one or two collapsed data disks, but it can also recover the data of three or more simultaneously collapsed data disks.

Key words: odd parity; data disk; redundancy technology; collapse; data recovery

如今, 大型数据中心的存储基础设施通常部署上万块硬盘, 随着存储阵列中硬盘数量的增多, 该阵列发生故障的概率在随之增加, 比如, 一个由 100 块硬盘构成的阵列中, 每块硬盘的 MTBF 大约为 750,000 小时, 则整个阵列每隔 7500 小时就可能会发生硬盘崩溃. 另外, 磁盘数量的增大还使得多个磁盘同时发生故障的可能性加大, 特别当某种灾难发生时这种可能性就更大. 因此, 数据恢复技术就成为当今数据管理技术中的一大热点话题^[1].

在数据中心, 对数据盘的备份可以采用镜像备份技术, 一旦数据盘或备份盘发生故障, 均可以采用镜像盘对数据进行恢复. 在此情形下, 只要损坏的磁盘数量只有一个均可以对数据进行恢复. 如果出现两个磁盘同时崩溃的现象, 只要不出现数据盘与对应的备份盘同时崩溃, 也可以恢复数据. 但在镜像备份中, 当数据盘与其镜像盘同时崩溃时其数据可能将无法恢复, 同样, 三个磁盘同时出现故障时也有类似的情形.

另外, 采用镜像备份时, 数据盘的数量和备份盘的数量应相等.

但是, 如果我们采用基于奇偶校验的冗余码技术, 在两个磁盘或者三个磁盘同时崩溃的情况下, 任何两个盘或者三个盘同时崩溃, 均可以对数据进行恢复. 而且, 通常冗余盘的数量低于数据盘数量.

1 基于奇偶校验的单盘崩溃数据恢复策略

以下分析中均以八位二进制代码表示磁盘中完整的数据块.

假设有两个任意八位二进制代码: 10110010、11010101, 对每个位进行异或操作进行奇偶校验, 将得到异或码 01100111^[2]. 事实上, 这三个码字互为异或, 第三个码字一定会在其它两个码字的基础上通过异或操作而得到, 或者说, 偶数个 1 校验值为 0, 奇数个 1 校验值为 1.

如果这三个码字代表三个不同的磁盘, 其中两个

^① 收稿时间:2013-01-21;收到修改稿时间:2013-03-07

为数据盘, 一个为冗余盘, 当其中的任意一个磁盘发生故障, 通过对其余两个磁盘的异或操作进行奇偶位校验, 就能够恢复被损坏盘的数据.

在此, 数据盘数量为 2, 冗余盘数量为 1.

2 两个磁盘同时崩溃后的数据恢复策略

当两个以上磁盘同时发生崩溃时, 不能简单地用异或操作对其进行数据恢复. 采用基于 RAID6 的冗余技术^[3], 首先建立一张冗余表, 如表 1. 根据冗余表反馈的 0 和 1 的分布规律来确定如何用未损盘数据对已损盘数据进行恢复.

表 1 修复双磁盘冗余码

磁盘编号	数据盘				冗余盘		
	1	2	3	4	5	6	7
冗余码	1	1	1	0	0	0	1
	1	1	0	1	0	1	0
	1	0	1	1	1	0	0

对于由 7 个磁盘组成的数据存储系统, 首先制作一张长度为 7 的冗余码表^[2], 此表有以下特征:

- 1) 每列均有 1;
- 2) 冗余盘部分每列一个 1;
- 3) 数据盘部分每列两个以上的 1;
- 4) 每行 1 的个数为偶数;
- 5) 7 列冗余码均不相同.

按偶数个 1 为 0 校验、奇数个 1 为 1 校验进行校验. 假设 1、2、3、4 号数据盘已分别写入以下数据:

1 号: 10110011、2 号: 01010101、3 号: 00100011、4 号: 11011011

按表 1 进行奇偶校验得到 5、6、7 号盘的数据为:

5 号: 01001011、6 号: 00111101、7 号: 11000101

其中, 5 号盘与 1、3、4 号盘进行异或操作; 6 号盘与 1、2、4 号盘进行异或操作; 7 号盘与 1、2、3 号盘进行异或操作. 每行均为四个 1, 形成偶校验.

如果 2、4 号盘同时崩溃, 可用冗余码第一行进行数据判断, 用 1、3、7 号盘的数据对 2 号盘进行数据恢复, 2 号盘的数据被恢复之后, 可用冗余码表第二行或第三行对 4 号盘进行数据恢复.

在此方案中, 数据盘数量为 4, 冗余盘数量为 3.

3 基于奇校验的三磁盘崩溃数据恢复策略

3.1 建立冗余表

假设系统包含 13 个磁盘, 其中数据盘数量为 8, 编号为 1、2、...8. 冗余盘数量为 5, 编号 9、10、...

13. 建立如表 2 的冗余表, 它有如下特征:

- 1) 每一列均有 1;
- 2) 冗余盘的每一列仅有一个 1;
- 3) 数据盘的每一列至少两个 1;
- 4) 数据盘中的每一行均为偶数个 1;
- 5) 13 个磁盘冗余码每行每列各不相同.

表 2 修复三磁盘崩溃冗余码

盘号 冗余码	数据盘								冗余盘				
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	0	0	1	0	1	1	1	0	0	0	0
2	0	1	1	1	0	0	0	1	0	1	0	0	0
3	1	0	1	0	0	1	0	1	0	0	1	0	0
4	0	1	0	1	1	0	1	0	0	0	0	1	0
5	0	1	1	0	1	1	0	0	0	0	0	0	1

3.2 冗余盘数据规则

数据盘上的数据是随机而任意的, 而冗余盘上的数据必须根据数据盘的内容进行设置.

八个数据盘的数据可以任意写入:

1 号: 10110010、2 号: 11001100、3 号: 00110101、4 号: 01010010、5 号: 11001100、6 号: 10101010、7 号: 11100010、8 号: 10011100

以表 2 为参考, 冗余盘中的数据按以下规则设置:

1) 9 号盘由 1、5、7、8 号数据盘进行奇校验, 每个盘对应位如果 1 的个数为偶数, 则 9 号盘的对应位为 0, 反之, 如果 1、5、7、8 号盘的对应位为奇数个 1 时, 9 号盘的对应位则为 1;

2) 10 号盘由 2、3、4、8 号数据盘进行奇校验;

3) 11 号盘由 1、3、6、8 号数据盘进行奇校验;

4) 12 号盘由 2、4、5、7 号数据盘进行奇校验;

5) 13 号盘由 2、3、5、6 号数据盘进行奇校验;

如此, 冗余表中的每一行有奇数个 1, 形成奇校验. 最终各冗余盘校验码为:

9 号: 00000000、10 号: 00110111、11 号: 10110001、12 号: 10110000、13 号: 10011111

3.3 写数据

当对数据盘重新写数据后, 相对应的冗余盘代码必须随之改变. 设 1 号数据盘变为 00101101, 如此, 9 号冗余盘和 11 号冗余盘的冗余码应分别变为 10011111 和 00101110, 其余冗余盘不变.

冗余盘代码修正的过程实际上是一个 CPU 查询冗余码表并进行相对简单的异或操作的过程, 由于不涉及其他的算术运算, 因此, CPU 消耗的时间资源较少, 冗余效率较高.

同理, 如果对 5 号数据盘重新写数据后, 则 9、12、13 号冗余盘也将随之做相应改变.

3.4 三个磁盘同时崩溃后的数据恢复

任何三个磁盘损坏后, 均可通过表 2 进行数据恢复, 前提是表 2 中三个盘的冗余码中仅有一个 1, 且每一列代码均不相同, 否则可能出现三个坏盘只能恢复其中一个的现象.

假如 2、3、7 号盘同时损坏, 由表 2 查询可知, 在冗余码中的第三行, 2 号盘的冗余码为 0, 3 号盘的冗余码为 1, 7 号的冗余码为 0, 符合“三个盘的冗余码中仅有一个 1”的要求.

首先用 1、6、8、11 号盘的八位代码对 3 号数据盘相应位进行奇校验, 以第三位为例, 由于 1、6、8、11 号 4 个盘的第三位有 3 个 1 和 1 个 0, 所以 3 号盘的第三位确认为 1, 等等. 最后 3 号盘该数据块的第三位被恢复为: 00110101. 3 号盘数据被恢复之后, 后续两个盘的数据恢复变得简单, 用表 2 冗余码的第 2 行或者第 5 行对 2 号盘进行数据恢复, 因为这两行的 2 号和 7 号盘的冗余码分别为 1 和 0. 但第 4 行却由于 2 号盘和 7 号盘冗余码同时为 1 而不能使用.

此例列举的实例中, 三个盘均为数据盘. 如果损坏盘中包含冗余盘, 或者损坏的三个盘均为冗余盘, 其数据恢复方法相同, 只要能保证冗余表中三个损坏盘的冗余码仅有一个为 1 而其余两个为 0 即可.

表 2 拥有 8 个数据盘, 而其中三个数据盘同时损坏的情况共有 56 种组合情形. 表 2 所列 0、1 序列可以做为公式使用, 它可以对 56 种所有可能损坏的盘序进行数据恢复. 以下 VB 代码证明了在表 2 中的任何一种情况均能保证三个损坏盘的冗余码中仅有一个 1, 即所有组合均成立.

```
Dim p(8) (定义 8 个磁盘的冗余码)
```

```
p(1) = "10100": p(2) = "01011"
```

```
p(3) = "01101": p(4) = "01010"
```

```
p(5) = "10011": p(6) = "00101"
```

```
p(7) = "10010": p(8) = "11100"
```

```
ki = 0
```

```
For i = 1 To 6 (损坏的某一个盘可能是 1-6 号盘)
```

```
For j = i + 1 To 7 (损坏的另一个盘可能是 2-7 号盘)
```

```
For k = j + 1 To 8 (损坏的第三个盘可能是 3-8 号盘)
```

```
For h = 1 To 5 (从第 1 位到第 5 位冗余码, 判断三个冗余码中是否仅有一个 1)
```

```
x=Mid(p(i),h, 1) (取第 1 个损坏盘第 h 位冗余码)
```

```
y=Mid(p(j),h, 1) (取第 2 个损坏盘第 h 位冗余码)
```

```
z=Mid(p(k),h,1) (取第 3 个损坏盘第 h 位冗余码)
```

```
If Val(x) + Val(y) + Val(z) = 1 Then (三个损坏盘的冗余码是否仅有一个 1)
```

```
ki = ki + 1 (56 种组合序号)
```

```
Print ki; ":"; i; ","; j; ","; k; ":"; x; ","; y; ","; z (该种组合能够恢复所有损毁盘的数据)
```

```
Exit For
```

```
End If
```

```
Next
```

```
If h = 6 Then
```

```
ki = ki + 1
```

```
Print ki; ":"; i; ","; j; ","; k (该种组合不成立, 不能够完全进行数据恢复)
```

```
End If
```

```
Next: Next: Next
```

以上代码并没有考虑冗余盘的存在, 但由于冗余盘的每一行仅有一个 1, 而且每一列均不相同, 它对组合可行性判断没有影响. 因此, 表 2 这种 0、1 序列具备完全可行性, 适合于每一种组合.

3.5 均衡性配置

以上操作中, 任何一个数据盘数据的改变均要涉及冗余盘的读写, 因此, 冗余盘的工作负荷要远远大于数据盘. 事实上, 无论是数据盘还是冗余盘, 它们之间都是互为异或(互为奇偶对应)^[2,4], 因此, 为了保持所有磁盘工作强度的均衡性, 可将冗余盘所有存储空间按一定规则均匀分布到全部 13 个磁盘^[5].

将 13 个磁盘分别命名为 n=0、1、2、…12 号, 设 F 为某一个磁盘的冗余柱面, F 除以 13 取余数 C, 即 $F \bmod 13 = C$, C 则表示某一数据盘的盘号.

当 F 分别取 0、13、26、39、…等数字作为磁盘柱面编号时, 余数 C=0, 因此, 0、13、26、39 等柱面将作为 0 号盘的基础柱面. 然后在基础柱面之上加 9 并上推四个柱面即 n+9、n+10、n+11、n+12, 这样, 对于 n=0 的基柱面, 与 9、10、11、12 共 5 个柱面作为 0 号磁盘的第一组冗余块; 同理, 对于 0 号磁盘的 n=13 的基柱面, 向上则与 22、23、24、25 形成另一组冗余块. 对 0 号盘的基柱面 26、39…等情况类推.

同理, 当 F 分别取 1、14、27、40、…等作为基柱面时, $C = F \bmod 13 = 1$. 将这些柱面作为 1 号盘的起始

冗余块,然后再将另外四个柱面: $n+9$ 、 $n+10$ 、 $n+11$ 、 $n+12$, 即 10、11、12、13 共 5 个柱面作为 1 号盘的第一组冗余块;第二组冗余块组合为 14、23、24、25、26 五个柱面. 等等.

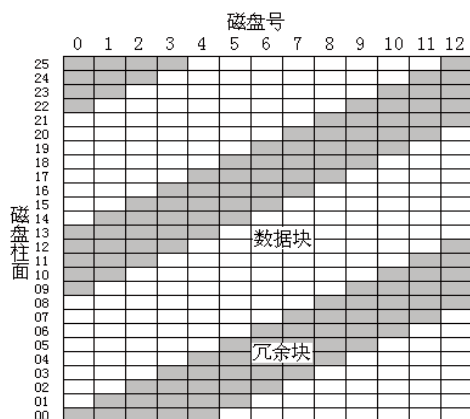


图 1 13 粒磁盘均衡性配置

2 号到 12 号盘的情况依此类推.

如此, 13 个盘被读和写的负荷概率均为 $1/13$. 具体应用时, 仍然采用表 2 作为冗余码表进行数据恢复操作.

由于每个盘除了被写正常数据外还要写冗余数据, 因此, 每个磁盘的写概率将大于 $1/13$.

仅对 8 个数据盘而言, 每个盘的被写概率是 $1/8$. 观察表 2 可知, 1、4、6、7 号盘的写操作将涉及两个冗余盘的写操作, 比如 1 号盘的写涉及 9、11 号冗余盘的写; 2、3、5、8 号盘的写将涉及三个冗余盘的写, 其中 2 号盘的写涉及 10、11、13 号冗余盘的写. 因此, 13 个盘总的写概率为 $(1/8) \times 3 \times 4 + (1/8) \times 4 \times 4 = 7/2$, 平均每个盘的写概率为 $(7/2)/13 = 7/26$.

4 扩展修复4个同时崩溃的磁盘

利用表 2 可以对某些情况下发生四个盘同时损坏的情况进行数据恢复, 前提是四个盘的冗余码表中仅有一个 1, 如 2、4、8、9 号盘同时崩溃, 利用表 2 中的第三行冗余码, 首先对 8 号盘利用 1、3、6、11 号盘的数据进行奇偶校验后进行数据恢复; 8 号盘被恢复之后, 再利用第一行冗余码对 9 号盘进行数据恢复; 最后再恢复 2、4 号盘.

对四个盘同时损坏的情况, 仅利用表 2 并不能对所有组合都能进行数据恢复, 如 2、4、5、7 号盘同时损坏则无法恢复, 它在表 2 中无法找到这四个盘有且仅有一个 1 的情形.

5 结语

目前, 单个磁盘的 MTBF 大约为 750000 小时, 则三个盘同时崩溃的概率大约为 63 万年, 13 个磁盘同时出现三个盘同时崩溃的概率为 $630000/56$, 约为 11000 年, 对于 1 个 1500 个磁盘的系统, 其概率大约为 100 年. 另据研究表明, 当一个磁盘损坏后, 其他磁盘损坏的概率将会上升^[1]. 这种概率对于一个大型数据存储机构而言是很高的. 特别当发生某种自然灾害如数据库节点的爆炸、火灾及恶意破坏的计算机病毒等, 此概率大增. 本文针对三个磁盘损毁设计的冗余码表, 可以以较少的 CPU 代价实现三个或更多磁盘同时崩溃后的数据恢复.

对于数据中心, 在实际应用过程中将每 13 个磁盘划分为一组, 就可以对存储系统中所有磁盘进行三磁盘损坏的恢复数据操作.

参考文献

- 1 董欢庆, 李战怀, 林伟. raid_vcr: 一种能够承受三个磁盘故障的 raid 结构. 计算机学报, 2006, 29(5): 792-800.
- 2 Garcia-Molina H, Ullman JD, Widom J. Database System Implementation. Palo Alto, California: Stanford University, 2010, 1: 584-587.
- 3 汪中夏, 张京生, 刘伟. RAID 数据恢复技术揭秘. 北京: 清华大学出版社, 2010. 35-45.
- 4 刘伟. 数据恢复技术深度揭秘. 北京: 电子工业出版社, 2010. 51-54.
- 5 张京生, 汪中夏, 刘伟. 数据恢复方法及案例分析. 北京: 电子工业出版社, 2008. 474-478.
- 6 戴士剑. 数据恢复与硬盘修理. 北京: 电子工业出版社, 2012. 4-7.
- 7 佚名. 解析 RAID6: 最新的冗余技术. 2006-10-07. <http://stor.51cto.com>.