第 26 卷 第 10 期
2003 年 10 月

计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS

Vol. 26 No. 10
Oct. 2003

容许两个盘故障的磁盘阵列数据布局与图分解的条件和存在性研究

周 杰[1)] 王 刚[2)] 刘晓光[2)] 刘 璟[2)]

[1)] (华南理工大学计算机科学与工程学院 广州 510641)

[2)] (南开大学信息科学技术学院 天津 300071)

摘 要 从一个新的途径讨论容许两个盘故障的磁盘阵列数据布局:把由数据单元和通过"异或"运算得到的校验单元组成的校验组用一个图表示,把校验组容许两个盘故障的阵列布局归结为校验组的单元集合的划分,进而转化为校验组的图的顶点和边组成集合的满足一定条件的分解.证明了校验组容许两个盘故障的单元集合划分的充分必要条件及存在性;讨论了优化阵列布局方案性能的条件;给出了阵列布局的步骤.从而为设计具有最优性能的容许两个盘故障的磁盘阵列数据布局方案提供了有效的途径.

关键词 校验组;校验组的 k-划分;校验组的图;可重构森林;闭路

中图法分类号 TP333

The Study of Graph Decompositions and Placement of Parity and Data to Tolerate Two Failures in Disk Arrays: Conditions and Existance

ZHOU Jie[1)] WANG Gang[2)] LIU Xiao-Guang[2)] Liu Jing[2)]

[1)] *(School of Computer Science and Engineering, South China University of Technology, Guangzhou* 510641*)*

[2)] *(College of Information Technology and Science, Nankai University, Tianjin* 300071*)*

Abstract A novel method for tolerating up to two disk failures in disk arrays has been presented. By representing a check group consisting of date and parity units with a graph, the conditions for tolerating two disk failures in disk arrays becomes to that of partitions of check group, and thus to that of the decompositions of its graph. A necessary and sufficient condition for the partition of check group is proved; the existence of the partition is given; the condition for optimizing the performance of the placement scheme is discussed; and the step of placement of the date and parity in a disk array is shown. It presents an efficient method for placement schemes with optimizing performance to tolerating two disk failures in disk arrays.

Keywords check group; k-partition of a check group; graph of check group; recoverable forest; close path

1 引 言

独立磁盘的冗余阵列 RAID (Redundant Array of Independent Disks) (简称磁盘阵列或阵列)[1]技术是解决 CPU 处理速度和输入/输出 (I/O) 性能之间发展不平衡的一个有效方案.但是,随着盘数的增多,阵列的可靠性随之降低[1,2].一般情况下,一个磁盘阵列的平均无故障时间 (MTTF) 为: $MTTF =$ 单个盘的 $MTTF$/阵列中盘的总数.此式表明,随着阵列中盘数的增多,阵列的平均无故障时间随之下降.

容许单个盘故障的磁盘阵列数据布局已有大量

收稿日期:2001-08-19;修改稿收到日期:2002-11-15.本课题得到国家自然科学基金(60273031)和高等学校博士学科点专项科研基金(2000005516,20020055021)资助.周 杰,男,1964 年生,博士,副教授,研究方向为组合优化、图论. E-mail: zjwqd@263.net.王 刚,男,1974 年生,讲师,主要研究方向为并行与分布式系统、海量存储技术.刘晓光,男,1974 年生,博士,研究方向为并行与分布式系统、算法分析、海量存储技术等.刘 璟,男,1942 年生,教授,博士生导师,研究方向为并行与分布式系统、算法分析、海量存储技术等.

的研究[1,3,4]. 容许多个盘同时故障,特别是容许两个盘同时故障的阵列布局也有一些结果[2,5~9]. Hellerstein 和 Gibson 给出的 2D 布局方案[5]及 Ng 给出的 Crosshatch 布局方案[6]等是把校验单元集中在校验盘上. 然而每次对用户数据的更新都需要更新校验单元,因此校验盘就成为阵列 I/O 的瓶颈. 而 Park 给出的 RM2 布局方案[7,8]、Blaum 等给出的 EVENODD 布局方案[2]以及最近由 Nam-Kyu Lee 等提出的 DH1 和 DH2 布局方案[9],把校验单元散布到阵列的每个盘中,以解决瓶颈问题. 但是,这些方案都有过多的小写额外开销,或者限定阵列盘数为素数.

在容许多个盘故障的磁盘阵列数据布局中,关键是如何实现"容许多磁盘故障"的条件以及如何使阵列布局方案的性能达到最优. 本文从一个新的途径讨论容许两个盘故障的磁盘阵列数据布局:把由数据单元和通过"异或"运算得到的校验单元组成的校验组的单元集合的划分作为阵列布局的一个主要步骤,使容许两个盘故障的条件通过校验组的单元集合的划分来实现. 为此,把校验组用一个图来表示,把校验组容许两个盘故障的单元集合划分转化为校验组的图的顶点和边组成集合的满足一定条件的分解. 证明了校验组容许两个盘故障的单元集合划分的充分必要条件及存在性;讨论了在冗余率达到最低时确定校验组的图的条件;给出了阵列布局的步骤. 用一个图表示一个校验组保证了在容许两个盘故障的条件下阵列布局方案有低的小写的额外开销. 为了降低编码和解码算法的复杂性,本文主要利用"异或(Exclusive-OR)"运算计算校验单元. 图论参考书见文献[10].

2 校验组和校验组的图

在磁盘阵列数据布局(简称阵列布局)中,用户数据被划分成大小相等的数据单元,依次编号为 0,1,2,…,称为用户数据单元序列. 若干相继数据单元组成一个数据组.

把一个数据组中的数据单元按某种方式组合,对每一个组合运用"异或"运算计算出校验单元,增加校验单元的数据组称为校验组(图 1). 数据单元和校验单元统称为单元.

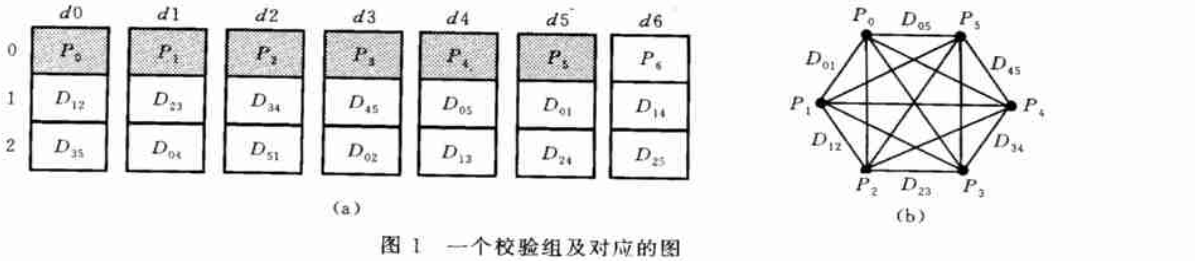

图 1 一个校验组及对应的图

在校验组中,一个校验单元由若干数据单元计算;同时,一个数据单元亦可被用来计算多个校验单元. 在容许两个盘故障的磁盘阵列数据布局中,每个数据单元必至少被用来计算两个校验单元(数据单元的镜像也看成是校验单元). 本文讨论校验组中每个数据单元恰好被用来计算两个校验单元的情形. 对这样的校验组可用一个无环图 G 直观地表示:校验单元对应 G 的顶点,数据单元对应 G 的边;数据单元 D 对应的边连接校验单元 P_i, P_j 对应的顶点,当且仅当 D 在校验组中被用来计算 P_i 和 P_j. 称图 G 为校验组的图.

设一个磁盘阵列含有 N 个盘,依次编号为 0,1,…,$(N-1)$. 一个盘上保存一个单元的空间称为一个物理单元. 每一个盘看成由一列物理单元组成,依次编号为 0,1,2,…:一个盘上的物理单元的标号称为偏移量.

例 1. 图 1(a)给出的是校验组$\{D_{ij} \mid 0 \le i \le 4, i < j \le 5\} \cup \{P_i \mid 0 \le i \le 5\}$的布局. 其中每一列代表一个盘,分别记为 $d0, d1, \cdots, d6$. 图(a)中最左边的一列数字代表各个盘上物理单元的偏移量. D_{ij} 是数据单元,P_i 是校验单元;D_{ij} 的下标表示这个数据单元被用来计算校验单元 P_i 和 P_j,校验单元 P_i 由下标含有 i 的数据单元通过"异或"运算得到,如 $P_0 = D_{01} \oplus D_{02} \oplus D_{03} \oplus D_{04} \oplus D_{05}$ 等. 图中所有数据单元组成一个数据组,所有单元组成一个校验组,其中每一个数据单元被用来计算两个校验单元,每一个校验单元由 5 个数据单元来计算.

图 1(b)是(a)的校验组的图,图的顶点和边的标号表示其对应的校验单元和数据单元(图中只标记了部分边). 这是一个有 6 个顶点的完全图 K_6.

由图 1 可以看出,用图表示一个校验组可以直观地反映出校验组中数据单元和校验单元之间的关系. 进一步,通过一个图还可以很容易地给出一个校验组. 下面以完全图 K_6 为例说明如何利用图来确定校验组.

例 2. 设完全图 K_6(图 1(b))的顶点集为$\{P_i \mid$

$0 \le i \le 5\}$,边集为$\{D_{ij} \mid 0 \le i \le 4, i < j \le 5\}$,这里$D_{ij}$连接顶点$P_i$,$P_j$,共有 15 条边. 指定用户数据单元序列中第 0 到第 14 号数据单元为一个数据组. 并且,给出数据组中数据单元与图的边集的一个一一对应,例如设第 0 到第 14 号数据单元分别对应边$D_{01}, D_{02}, \cdots, D_{05}, D_{12}, \cdots, D_{15}, \cdots, D_{45}$. 最后,对图中每个顶点关联的边对应的数据单元运用"异或"运算计算出对应的校验单元,这样就得到一个校验组,仍记为$\{D_{ij} \mid 0 \le i \le 4, i < j \le 5\} \cup \{P_i \mid 0 \le i \le 5\}$. 这个校验组的图就是完全图 K_6.

3　容许两个盘故障的阵列布局条件

3.1　校验组的布局与单元集合划分

确定校验组后,磁盘阵列的数据布局就是把校验组的每一个单元分配到阵列的物理单元中,称为校验组的布局. 给定校验组的一个布局,被分配到每一个盘上的单元是该校验组单元集合的一个子集. 从而,校验组的布局对应校验组的单元集合的划分. 进一步,校验组的单元集合的划分对应校验组的图的顶点和边所成集合的分解. 基于这样的事实,给出如下一些定义.

定义 1.　若校验组的一个单元子集中的单元都可由该子集以外的单元通过"异或"运算计算出,则称这个单元子集为可重构单元子集,否则称为不可重构单元子集.

定义 2.　(1) 对校验组的一个布局,若任意 k 个盘上的单元组成的单元子集是可重构单元子集,则称为校验组的容许 k 个盘故障的布局,简称校验组的 k-布局.

(2) 一个阵列布局能容许 k 个盘故障是指,当阵列中任意 k 个盘故障时,其上的每一个单元都可通过无故障盘上的单元重构.

(3) 若校验组的单元集合的划分中任意 k 个单元子集的并集是可重构单元子集,则称为校验组的容许 k 个盘故障的单元集合划分,简称校验组的 k-划分(定义中的 k 均指具有所述性质的极大者).

下面是几个与图有关的概念. 注意,这些概念比图论中相应概念包含的范围更广.

定义 3.　子图:一个图的顶点和边的任意子集均称为子图. 路(圈、树、森林、连通子图):图论中的路(圈、树、森林、连通子图)以及在其上去掉一些顶点(不去掉边)剩下的子图均称为路(圈、树、森林、连通子图).

由定义 3 知,校验组的单元子集在校验组的图中对应一个子图,称为单元子集的子图. 进一步,校验组的单元集合的划分对应校验组的图的子图分解.

定义 4.　对校验组的图的子图分解,若任意 k 个子图的并在校验组中都对应一可重构单元子集,则称为校验组的图的容许 k 个盘故障的子图分解,简称 k-分解.

总结以上的分析和定义有:

定理 1.　校验组的 k-布局给出它的 k-划分,从而对应校验组的图的 k-分解;反之,校验组的图的 k-分解给出校验组的 k-划分,进而,如果不同子集存入不同盘,则给出此校验组的 k-布局. 一个阵列布局能容许 k 个盘故障当且仅当每个校验组的布局至少是 k-布局.

由定理 1,只需以一个校验组为例来讨论磁盘阵列的布局. 为了给出校验组的 k-布局,只需给出它的 k-划分,进而只需给出校验组的图的 k-分解.

3.2　校验组的 2-划分条件和存在性

下面利用校验组图的子图结构给出校验组的 2-划分条件. 首先考察校验组的可重构单元子集的子图的结构及重构算法.

例 3.　对图 1 (a) 的布局,假设第 0 号和第 1 号盘故障,其上的单元子集为$\{P_0, D_{12}, D_{35}, P_1, D_{23}, D_{04}\}$,对应的子图如图 2. 由此,容易得到故障盘上单元的重构算法:$D_{04} = P_4 \oplus D_{14} \oplus D_{24} \oplus D_{34} \oplus D_{45}$, $P_0 = D_{01} \oplus D_{02} \oplus D_{03} \oplus D_{04} \oplus D_{05}$;类似的可依次计算 $D_{35}, D_{23}, D_{12}, P_1$. 事实上,在图 1 (a) 的布局中的任意两个盘故障时,故障盘上的单元都可用类似的方法重构.

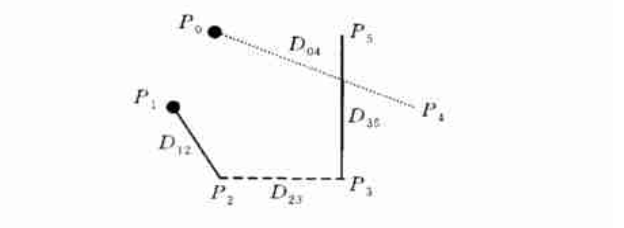

图 2　图1的布局中第0,1号盘上单元对应的子图

注意到单元子集$\{P_0, D_{04}\}$和$\{P_1, D_{12}, D_{23}, D_{35}\}$的子图的结构,有如下定义.

定义 5.　最多包含一个顶点的树称为可重构树;可重构森林是指每个分支都是可重构树的森林.

如图 2 给出的是由两棵可重构树构成的可重构森林. 由例 3,容易用归纳法证明.

引理 1.　若校验组的单元子集的子图是可重构森林,则这个单元子集为可重构单元子集.

其次,考察校验组的不可重构单元子集的子图的结构. 为此,对例 2 的由完全图 K_6 确定的校验组

给出一个布局如图 3(a).

例 4. (1) 对图 3(a)的布局,假设第 0,1 号盘故障,其上的单元子集为$\{P_0, D_{15}, D_{24}, P_1, D_{04}, D_{25}\}$,对应的子图如图 5(b). 下面说明这个单元子集为不可重构单元子集. 由完全图 K_6及其确定的校验组知,与此单元子集中的单元相关的计算校验单元的等式为

$$P_0 \oplus D_{04} = D_{01} \oplus D_{02} \oplus D_{03} \oplus D_{05},$$
$$D_{04} \oplus D_{24} = P_4 \oplus D_{14} \oplus D_{34} \oplus D_{45},$$
$$D_{24} \oplus D_{25} = P_2 \oplus D_{02} \oplus D_{12} \oplus D_{23},$$
$$D_{15} \oplus D_{25} = P_5 \oplus D_{05} \oplus D_{35} \oplus D_{45},$$
$$D_{15} \oplus P_1 = D_{01} \oplus D_{12} \oplus D_{13} \oplus D_{14}.$$

上面每个等式中等号右边的单元均在无故障盘上,因而是已知的,等号左边为未知量. 这是有 6 个未知量,5 个方程的方程组,所以没有唯一解. 因此,单元子集 $P_0, D_{15}, D_{24}, P_1, D_{04}, D_{25}$是不可重构单元子集.

(2) 当图 3(a)的布局中第 2,3 号盘故障时,其上的单元子集为$\{D_{01}, D_{12}, D_{23}, D_{34}, D_{45}, D_{05}\}$,对应的子图为图 3(c),有如下的方程组:

$$D_{01} \oplus D_{05} = P_0 \oplus D_{02} \oplus D_{03} \oplus D_{04},$$
$$D_{01} \oplus D_{12} = P_1 \oplus D_{13} \oplus D_{14} \oplus D_{15},$$
$$D_{12} \oplus D_{23} = P_2 \oplus D_{02} \oplus D_{24} \oplus D_{25},$$
$$D_{23} \oplus D_{34} = P_3 \oplus D_{03} \oplus D_{13} \oplus D_{35},$$
$$D_{34} \oplus D_{45} = P_4 \oplus D_{04} \oplus D_{14} \oplus D_{24},$$
$$D_{45} \oplus D_{05} = P_5 \oplus D_{15} \oplus D_{25} \oplus D_{35}.$$

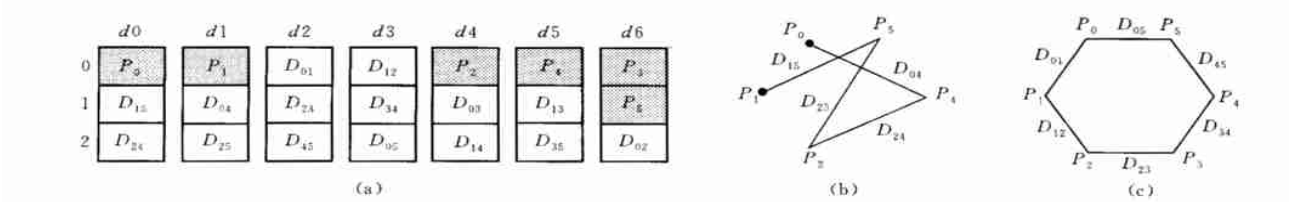

图 3 完全图 K_6对应的校验组的一个布局及故障盘上单元对应的子图

上面各等式中等号右边的单元均在无故障盘上,等号左边为未知单元. 这是有 6 个未知量,6 个方程的方程组,系数行列式为

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{vmatrix} = 0 \text{(在域 } F_2 \text{ 中)},$$

即这个方程组没有唯一解. 因而,单元子集$\{D_{01}, D_{12}, D_{23}, D_{34}, D_{45}, D_{05}\}$是不可重构单元子集.

注意到单元子集$\{P_0, D_{15}, D_{24}, P_1, D_{04}, D_{25}\}$和$\{D_{01}, D_{12}, D_{23}, D_{34}, D_{45}, D_{05}\}$的子图的结构,有下面定义.

定义 6. 一个图的如下任一子图称为闭路:

(1) 只包含两个端点不含中间点且至少含一条边的路;

(2) 只含边不含顶点的至少有两条边的圈.

如图 3 的(b),(c)均是完全图 K_6的闭路. 用与例 4 类似的方法可证下面引理.

引理 2. 校验组的图的闭路对应的单元子集是不可重构单元子集.

下面的引理 3 证明了无环图的子图是可重构森林与其不包含闭路是等价的.

引理 3. 无环图 G 的子图 H 是可重构森林当且仅当 H 不包含闭路.

证明. 若 H 是可重构森林,由定义 5 和 6 知,H 不含闭路. 反之,设 H 不含闭路. 由定义 6 的(2),H 不含圈,从而是森林. 若 H 中有一个连通支(是一棵树)包含两个顶点,由定义 6 的(1),这个连通支包含闭路. 因此 H 的每个连通支都是最多含一个顶点的树,即为可重构树. 所以 H 是可重构森林. 证毕.

假设校验组的图是无环图,即验组中每个数据单元都被用来计算两个校验单元. 结合引理 1,2 和 3 有下面定理.

定理 2. 设 B 是校验组的一个单元子集,则下面 3 个论断是等价的:

(1) B 是可重构单元子集;

(2) B 的子图不含闭路;

(3) B 的子图是可重构森林.

推论 1. 校验组的单元集合划分最多为 2-划分. 特别地,若校验组的图含有重边,则它的任意单元集合划分均是 1-划分.

证明. 校验组的图中任一边及关联的顶点构成一条闭路. 此闭路对应的三个单元构成一个不可重构单元子集. 对校验组的任意单元集合划分,至多有 3 个子集包含这 3 个单元,并且包含这 3 个单元的单元子集的并集是一不可重构单元子集. 于是校验组的单元集合划分最多为 2-划分.

当校验组的图含有重边时,连接同一对顶点的两条重边构成一条闭路,对应校验组的一个不可重构单元子集. 对校验组的任意单元集合划分,至多有两个子集包含这两个单元. 证毕.

由推论 1,在构造容许两个盘故障的阵列数据

布局时,应选择简单图(不含环和重边)作为校验组的图,这样才能保证校验组存在 2-划分. 因此,下面均假设校验组的图为简单图.

再由定义 2,定义 4 和引理 3 有下面推论.

推论 2.　对校验组的单元集合的一个划分,下面 3 个论断是等价的:

(1) 此划分是 2-划分;

(2) 此划分中任两个单元子集的并集的子图不含闭路;

(3) 此划分中任两个单元子集的并集的子图是可重构森林.

推论 3.　对校验组的图的一个子图分解,下面 3 个论断等价:

(1) 此分解是 2-分解;

(2) 此分解中任意两个子图的并不含闭路;

(3) 此分解中任意两个子图的并是可重构森林.

推论 2 和 3 给出了校验组(校验组的图)的 2-划分(2-分解)准则. 下面定理说明,对任一校验组(校验组的图),其 2-划分(2-分解)都是存在的. 事实上,设一个校验组的图是简单图,把图的每个顶点和每条边作为一个子图,则给出校验组的图的一种 2-分解,称为平凡分解;相应的给出校验组的一个 2-划分,称为平凡划分.

定理 3. 由简单图确定的校验组都存在 2-划分.

校验组的平凡划分给出的阵列布局仅满足容许两个盘故障的条件. 在磁盘阵列的数据布局中,不仅仅要求一个布局方案满足"容许两个盘故障"的条件,同时还希望阵列的其它性能(如冗余率、小写额外开销)达到最优. 下一节将讨论在容许两个盘故障的阵列布局中,如何选择校验组的图使得校验组的 2-划分给出的阵列布局的性能达到最优.

定理 4.　若校验组有 n 个校验单元(即校验组的图有 n 个顶点),则校验组的一个可重构单元子集至多包含 n 个单元. 从而若一个校验组含有 m 个数据单元 n 个校验单元,则校验组的 2-划分中至少要分成 $2(n+m)/n$ 个子集,即校验组的容许两个盘故障的布局至少需要 $2(n+m)/n$ 个盘.

证明.　对于有 n 个校验单元的校验组,在重构一个单元子集中的单元时,至多有 n 个等式(一个校验单元对应一个等式,参见例 3,例 4). 这样,对单元数多于 n 的单元子集,在重构时,就会得到一个至多有 n 个方程,而未知量的个数大于 n 的方程组. 因此,这样的单元子集是不可重构单元子集.

校验组的 2-划分要求任意两个子集的并构成一个不可重构单元子集,从而有 m 个数据单元 n 个校验单元的校验组的 2-划分至少要分成 $2(n+m)/n$ 个子集.　　证毕.

最后再给出几个图的 2-分解的例子.

例 5. 4 个顶点的完全图 K_4(图 4(a))的分成 5 个子集的一种 2-分解为$\{P_0, D_{12}\}$;$\{P_1, D_{23}\}$,$\{P_2, D_{30}\}$,$\{P_3, D_{01}\}$,$\{D_{02}, D_{13}\}$. 这里 D_{ij} 连接顶点 P_i,P_j,下同.

5 个顶点的完全图 K_5(图 4(b))的分成 7 个子集的 2-分解为$\{P_0, D_{12}\}$,$\{P_1, D_{23}, D_{04}\}$,$\{P_2, D_{34}\}$,$\{P_3, D_{02}\}$,$\{P_4, D_{13}\}$,$\{D_{01}, D_{24}\}$,$\{D_{03}, D_{14}\}$.

6 个顶点的完全图 K_6(图 4(c))的分成 7 个子集的一种 2-分解为$\{P_0, D_{12}, D_{35}\}$,$\{P_1, D_{23}, D_{40}\}$,$\{P_2, D_{34}, D_{51}\}$,$\{P_3, D_{45}, D_{02}\}$,$\{P_4, D_{50}, D_{13}\}$,$\{P_5, D_{01}, D_{24}\}$,$\{D_{03}, D_{14}, D_{25}\}$.

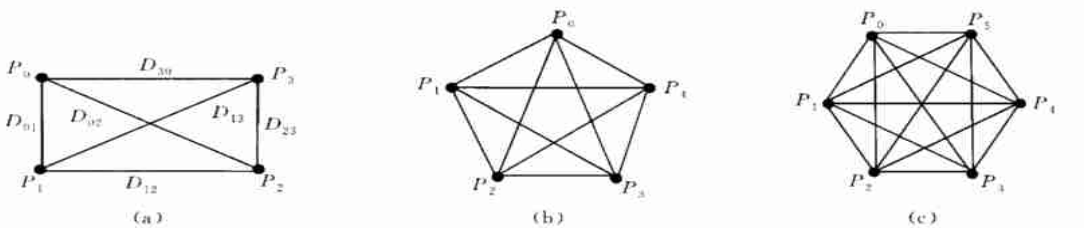

图 4　完全图 K_4,K_5,K_6

4　布局方案性能的优化

首先考查冗余率. 冗余率是描述阵列布局方案性能的主要参数之一. 校验组的冗余率定义为校验组的校验单元数与全部单元数的比值. 若一个校验组含有 n 个校验单元,m 个数据单元(即校验组的图含有 n 个顶点 m 条边),则校验组的冗余率为 $n/(n+m)$.

一个阵列布局方案中校验单元数与总单元数的比值称为此布局方案的冗余率. 阵列布局中,在保证可靠性及 I/O 性能的前提下,总希望布局方案的冗余率尽可能小. 根据编码理论中的 singleton bound 定理[11],对含有 N 个盘的阵列,它的容许 k 个盘故障的布局方案的冗余率至少为 k/N.

命题 1.　对一个阵列布局方案,若所有校验组

都含有 n 个校验单元, m 个数据单元,即所有校验组的冗余率均为 $n/(n+m)$. 则该布局方案的冗余率亦为 $n/(n+m)$.

下面考查对于给定阵列的盘数 N,如何选择校验组(即如何确定校验组的图),使得所确定的容许两个盘故障的阵列布局方案的冗余率为 $2/N$. 这里假设在阵列布局过程中,所有校验组的图都同构. 为此,由命题 1,只以一个校验组为例讨论.

定理 5. 给定阵列盘数 N,设校验组含有 n 个校验单元, m 个数据单元(即校验组的图含有 n 个顶点 m 条边),则校验组存在到阵列的容许两个盘故障的冗余率是 $2/N$ 的布局,当且仅当 $n/(n+m)=2/N$ 且校验组有一个分成 $N=2(n+m)/n$ 个子集的 2-划分(即校验组的图有一个分成 $N=2(n+m)/n$ 个子图的 2-分解)及 $n \geq N-1$.

证明. 由 $n/(n+m)=2/N$ 知,校验组的冗余率为 $2/N$;再由校验组有一个分成 $N=2(n+m)/n$ 个子集的 2-划分,根据定理 1,这种单元集合划分给出此校验组的容许两个盘故障的布局. 反之,若校验组的冗余率是 $2/N$,则有 $n/(n+m)=2/N$;再由定理 1,校验组到阵列的容许两个盘故障的布局给出它的容许两个盘故障的单元集合划分. 设该单元集合划分分成 t 个单元子集,则 $t \leq N=2(n+m)/n$. 由定理 4 有 $t \geq 2(n+m)/n=N$,从而 $t=N=2(n+m)/n$,即校验组有一个 2-划分,分得的单元子集数为 $N=2(n+m)/n$.

最后,由 $n/(n+m)=2/N$ 得 $m=n(N-2)/2$. 根据推论 1,校验组的图应为简单图. 从而, $m \leq n(n-1)/2$,即 $n(N-2)/2 \leq n(n-1)/2$,解得 $n \geq N-1$.

证毕.

由定理 5,对于给定的有 N 个盘的阵列,要构造一容许两个盘故障且冗余率为 $2/N$ 的布局方案,需选择同时满足如下条件的简单图 G 作为校验组的图:

(1) G 的顶点数 $n \geq N-1$;

(2) G 的边数 $m=n(N-2)/2$;

(3) G 有一个分成 N 个子图的 2-分解.

满足条件(1)和(2)的图是很多的. 特别地,当 $n=N-1$ 时, $m=(N-1)(N-2)/2$. 此时图 G 是有 $N-1$ 个顶点的完全图 K_{N-1},这个图给出的是满足条件"容许两个盘故障且冗余率是 $2/N$"的具有单元数最少的校验组. 在磁盘阵列的数据布局方案中,校验组的单元数越少,布局方案的 I/O 性能越好. 因此,希望对有 $N-1$ 个顶点的完全图 K_{N-1} 给出分成 N 个子图的 2-分解的方法. 目前还不清楚哪些完全图存在这种 2-分解以及如何分解.

例 5 给出的是有 4,5,6 个顶点的完全图的 2-分解. 其中对 K_4 和 K_6 的分解是满足如上条件的 2-分解. 可以证明完全图 K_5 不存在分成 6 个子图的满足条件的 2-分解.

如果找出完全图的满足如上条件的 2-分解有困难或者这种 2-分解不存在,应适当增加图的顶点数,如取 $n=N, N+1, \cdots$(相应地也增加校验组的图的边数),然后对相应的图寻找满足条件的 2-分解. 对每个这样的 n,有 $m=n(N-2)/2$ 条边的简单图是很多的. 因此对某个 n,比如 $n=N$,构造一个有 n 个顶点, $m=n(N-2)/2$ 条边的图,使其存在分成 N 个子图的 2-分解应该是可行的,这也是我们进一步的工作. 关于阵列数据布局的具体步骤将在下一部分讨论.

最后讨论阵列布局方案的小写额外开销. 在磁盘阵列中,当一次写入的数据小于或等于一个数据单元时,称为小写(small writes). 小写时,由于要修改相应的校验单元,由此而带来的过多的额外开销会降低阵列的吞吐量,从而会影响整个系统的性能. 例如,对于前面的由图所确定的校验组(每个数据单元恰好被用来计算两个校验单元)给出的阵列布局方案,当更新一个数据单元时,需先读出此单元的旧数据及与此数据单元有关的两个旧的校验单元(预读),然后利用"异或"运算分别计算出新的校验单元(新的校验单元 = 旧的数据单元 ⊕ 旧的校验单元 ⊕ 新的数据单元),最后写入新的数据单元及新的校验单元. 因此,每一次小写需要 3 次读和 3 次写,共 6 次 I/O 操作和两次计算,称为小写额外开销. 小写问题一直是磁盘阵列研究要考虑的一个重要问题. 在磁盘阵列数据布局选择校验组时,为了使所给出的布局方案有低的小写额外开销,应使得每个数据单元被用来计算尽可能少的校验单元.

在容许两个盘故障的阵列数据布局中,具有最低小写额外开销的阵列布局方案是对每个数据单元采用两次镜像(小写额外开销只有 3 次写操作),然而如此给出的布局方案具有最大的冗余率;其次是采用镜像和校验相结合(每次小写需要两次读和 3 次写共 5 次 I/O 操作和一次计算)[12],如此方法给出的布局方案的冗余率也不能达到最优. EVENODD 布局方案[2]的冗余率达到最优($2/N$),然而对该布局方案对角线上数据单元的更新却需要 $2(N-1)$ 次 I/O 操作和 $N-2$ 次计算;而 DH1 布局方案[9],更新一个数据单元需要 8 次 I/O 操作和 3 次计算(事实上,DH1 和 EVENODD 布局方案的校验组对应的是超图). 由前面段的说明,用本文的方法给出的布局方案,每一次小写只需要 3 次读和 3 次写共 6 次 I/

O 操作和两次计算. 因此,按照本文的方法给出的布局方案,不但可使得冗余率达到最优,而且还会使得小写额外开销相对较低.

5　磁盘阵列容许两个盘故障的数据布局的步骤

给定磁盘阵列的盘数 N(依次标号为 0,1, …, $(N-1)$),假设在阵列布局过程中,所有校验组的图都同构. 根据前面几节的讨论,可按以下步骤给出阵列布局:

(1) 给出校验组的图及其 2-分解:按照定理 5 后面的讨论,根据阵列的盘数 N 以及对阵列布局方案性能的要求,构造校验组的图,同时根据推论 2 给出该图的一个 2-分解(要求图的 2-分解所分得的子图数 $l \le N$).

(2) 确定校验组及其单元集合划分:按照给出的校验组的图的边数,把用户数据单元序列划分为数据组,指定数据组的数据单元集合与图的边集合的对应关系,再按照例 2 的方法确定每一个校验组. 同时由校验组的图的 2-分解对应校验组的 2-划分.

(3) 给出映射函数,即对每个校验组的单元集合划分的每一个子集指定存入阵列的盘号及其中单元在相应盘中的偏移量. 这里要求每一校验组的不同的子集对应不同的盘号,同一子集中的不同单元对应的偏移量也不同.

(4) 给出重构算法. 即给出当阵列中任意一个盘或两个盘故障时,对故障盘上单元重构的算法.

通过以上步骤实现的磁盘阵列数据布局称为磁盘阵列的一个数据布局方案,简称阵列布局方案.

运用以上步骤给出阵列布局方案,关键是确定校验组的图. 因为,校验组的图决定阵列布局方案的性能,决定校验组的单元子集是否存在满足条件的 2-划分,同时也影响映射函数和重构算法的确定. 另外,在确定映射函数时,一方面要考虑校验单元在阵列中各个盘上的均匀分布,同时也要考虑是否容易给出重构算法.

下面以一个例子来说明如上步骤.

例 6.　假设每个数据组包含 6 个数据单元,且对应的图均为完全图 K_4. 校验组的 2-划分由例 5 中对完全图 K_4 的分成 5 个子集的 2-分解给出. 由此确定的阵列布局方案如图 5(只列出 3 个校验组的布局). 这里阵列盘数 $N=5$, P_i^j 表示第 j 个校验组的第 i 个校验单元, $D_{i\,k}^j$ 表示第 j 个校验组中图 K_4 的边 $D_{i\,k}$ 对应的数据单元,即用来计算校验单元 P_i^j 和 P_k^j 的数据单元. 在这个布局方案中,为了使校验单元均匀分布到阵列的各个盘上,把每个校验组单元集合划分中的不含校验单元的子集存入到不同的盘上,如图第 0 个校验组的不含校验单元的子集放到第 4 号盘,第 1 个校验组的不含校验单元的子集放到第 3 号盘. 这样,每 5 个校验组作为一次循环.

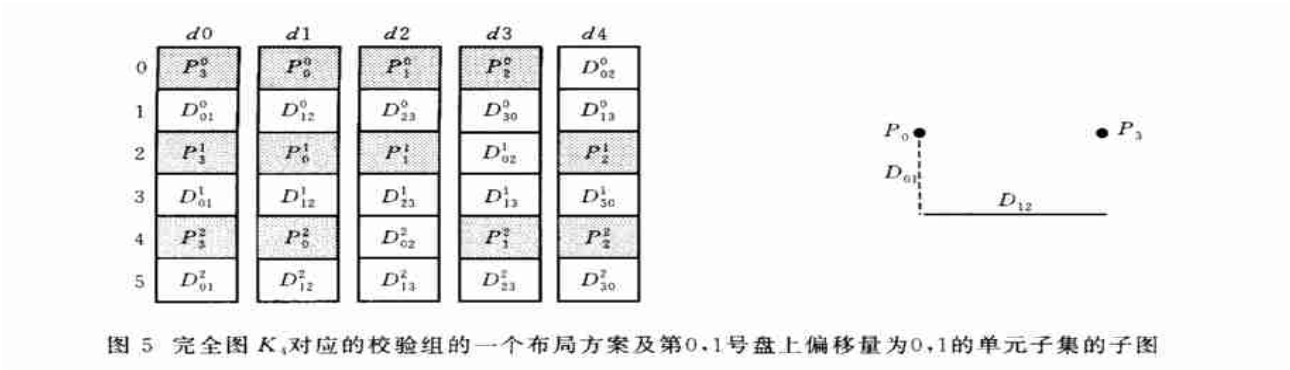

图 5　完全图 K_4 对应的校验组的一个布局方案及第0,1号盘上偏移量为0,1的单元子集的子图

下面给出这个布局的映射函数. 首先,用户数据单元依次编号为 0,1,2, …. 第 j 个校验组的数据单元按 D_{01}^j, D_{02}^j, D_{30}^j, D_{12}^j, D_{13}^j, D_{23}^j 的顺序依次标号为 0,1,2,3,4,5;也就是,当 $i=0$ 时,数据单元 $D_{i\,k}^j$ 是第 j 个校验组的第 $k-1$ 号数据单元,当 $i=1,2$ 时, $D_{i\,k}^j$ 是第 $i+k$ 号数据单元. 第 l 号用户数据单元是第 $\left\lfloor \frac{l}{6} \right\rfloor$ 个校验组的第 $l \bmod 5$ 号数据单元.

表 1 给出校验单元和数据单元存入阵列的盘号和偏移量.

最后说明对故障盘上单元重构的算法. 当一个盘故障时,如第 0 号盘故障,重构其上偏移量为 0 的单元. 由映射函数可确定此物理单元内的单元是 P_3^0,由校验组的图可知: $P_3^0 = D_{23}^0 \oplus D_{30}^0 \oplus D_{13}^0$;再由映射函数知数据单元 D_{23}^0, D_{30}^0, D_{13}^0 分别在第 2,3,4 号盘的偏移量为 1 的物理单元内,进而可重构 P_3^0.

表　1

单元	存入阵列的盘号	在相应盘上的偏移量
P_i^j	$(i+1) \bmod 4 + \left\lfloor \frac{(i+1) \bmod 4 + j \bmod 5}{4} \right\rfloor$	$2j$
$D_{i\,k}^j$ $i=0,k=2$ 或 $i=1,k=3$	$4-j \bmod 5$	$2j+i$
其它	$i + \left\lfloor \frac{(i+1)+j \bmod 5}{4} \right\rfloor$	$2j+1$

当两个盘故障时,如第 0 号和第 1 号盘故障,重构其上偏移量为 0 和 1 的单元. 由映射函数可知对应的单元子集为{ P_3^0, D_{01}^0, P_0^0, D_{12}^0},其对应的子图如图 5. 从而有 $P_3^0 = D_{23}^0 \oplus D_{30}^0 \oplus D_{13}^0$, $D_{12}^0 = P_2^0 \oplus D_{23}^0 \oplus D_{02}^0$, $D_{01}^0 = P_1^0 \oplus D_{12}^0 \oplus D_{13}^0$, $P_0^0 = D_{01}^0 \oplus D_{02}^0 \oplus D_{30}^0$. 再由映射函数可确定上面等式中等号右边单元所在物理单元的盘号和偏移量,进而可重构单元 P_3^0, D_{01}^0, P_0^0, D_{12}^0.

6 结 论

本文从一个新的途径讨论容许两个盘故障的阵列布局:把校验组用一个图表示,校验组的容许两个盘故障的布局归结为它的容许两个盘故障的单元集合划分,进而转化为校验组的图的顶点和边组成集合的容许两个盘故障的分解. 为设计容许两个盘故障的阵列布局提供了一种有效的方法. 用图表示一个校验组不但可以很容易地给出一个校验组,而且可以直观地反映出校验组中数据单元和校验单元之间的关系,同时把校验组的容许两个盘故障的阵列布局转化为校验组的图的 2-分解这样一个数学问题. 按照本文的方法给出的布局方案,不但可使得阵列布局方案的冗余率达到最优,而且还会使得小写额外开效相对较低. 我们是从总体上考虑磁盘阵列的数据布局,由此,已知的布局方案(如 RM2, DH2 等)只是某类校验组的特殊的单元集合划分确定的布局方案,也就是某类特殊的图的特定分解对应的布局方案.

本文假定校验组中每个数据单元只被用来计算两个校验单元,从而只讨论容许两个盘故障的阵列布局. 对每个数据单元被用来计算多个校验单元的情况以及容许多个盘故障的阵列布局是我们进一步讨论的问题.

参 考 文 献

1 Katz R H, Gibson G A, Patterson D A. Disk system architecture for high performance computing. IEEE Proceedings, 1989, 77(12):1842～1858

2 Blaum M, Brady J, Bruck J, Menon J. EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures. IEEE Transactions on Computing, 1995, 44(2): 192～202

3 Lee E K, Katz R H. The performance of parity placements in disk arrays. IEEE Transactions on Computing, 1993, 42(6): 651～664

4 Thomas J E Schwarz, Steinberg J, Burkhard W A. Permutation Development Data Layout (PDDL) disk array declustering. In: Proceedings of HPCA '99, Orlando, 1999. 214～217

5 Hellerstein L, Gibson G A, Karp R M, Patterson D A. Coding techniques for handling failures in large disk arrays. Algorithmica, 1994, 12(3-4): 182～208

6 Ng S W. Crosshatch disk array for improved reliability and performance. In: Proceedings of International Symposium on Computer Architecture, Chicago, 1994. 255～264

7 Park C. Efficient placement of parity and data to tolerate two disk failures in disk array systems. IEEE Transactions on Parallel and Distribute Systems, 1995, 6(11): 1177～1184

8 Park C, Choe T Y. Striping in disk arrays RM2 enabling the tolerance of double disk failures. Http://www.supercomp.org/sc96/proceedings/sc96PROC/CIPARK/INDEX.HTM

9 Lee N K, Yang S B, Lee K W. Efficient parity placement schemes for tolerating up to two disk failures in disk arrays. Journal of Systems Architecture, 2000, 46(15): 1383～1402

10 Bondy J A, Murty U S R. Graph Theory With Applications. London: The Macmillan Press Ltd., 1976

11 MacWilliams F J, Sloane N J A. The Theory of Error-correcting Codes. Amsterdam: North-Holland, 1977

12 Zhou Jie, Liu Xiao-Guang, Wang Gang, Liu Jing. Data parity and mirror layout for tolerating two disk failures in disk array. Computer Engineering and Applications, 2002, 38(18): 82～85 (in Chinese)
(周 杰,刘晓光,王 刚,刘 璟. 基于镜像和奇偶校验容许两个盘故障的磁盘阵列数据布局. 计算机工程与应用, 2002, 38(18):82～85)

ZHOU Jie, born in 1964, Ph. D., associate professor. His research interests include combinatorial optimization, graph theory, and computer networking.

WANG Gang, born in 1974, lecturer. His research interests include parallel and distributed system, mass storage technology.

LIU Xiao-Guang, born in 1974, Ph. D., postdoctoral fellow. His research interests include parallel and distributed system, mass storage technology.

LIU Jing, born in 1942, professor, Ph. D. supervisor. His research interests include algorithm analysis, parallel and distributed system, mass storage technology, and so on.