用蒙特卡罗方法和图描述法分析存储系统多容错编码可靠性*

王 刚[1+],葛广贺[1], 邓万禧[1], 刘晓光[1], 刘 璟[1]

[1](南开大学 信息技术科学学院, 天津市 300071)

Analyzing Reliability of Multi-Erasure-Correcting Codes of Storage Systems Using Monte Carlo Method and Graph Representation

Wang Gang[1+],Ge Guanghe[1], De Wanxi[1], Liu Xiaoguang[1], Liu Jing[1]

[1](Department of Computer Science, Nankai University, Tianjin City 300071, China)

+ Corresponding author: Wang Gang Phn: +86-22-2350-4780, Fax: +86-22-2350-4780, E-mail: wgzwp@163.com

Received 2004-00-00; Accepted 2004-00-00

Abstract: A simulation algorithm for reliability of multi-erasure-correcting codes for storage systems based on sequential Monte Carlo model is designed. Compared with traditional Markov model solving, the new method can analyzes not only simple MDS codes, but also complex fault tolerance of non-MDS codes, complex failure types, and non exponential distributed failure rate. An unrecoverable-erasure detecting algorithm for multi-erasure-correcting codes based on graph theory is also designed, this algorithm improves the plain algorithm in complexity greatly - from $O(2^n)$ to O(n). The program based on the new algorithms is implemented. The results of simulations show that, the new method improves the accuracy and the efficiency of multi-erasure-correcting codes reliability analysis indeed.

Key words: multi-erasure-correcting codes; reliability; Markov model; sequential Monte Carlo model; graph representation

摘 要: 本文设计了一种利用顺序蒙特卡罗仿真模型分析存储系统多容错编码可靠性的方法，与传统的求解 Markov 模型的方法相比，新方法不仅能对 MDS 码进行分析，对非 MDS 码的复杂容错特性、多种故障因素和不同故障概率分布均能很好地进行仿真。本文还提出了基于图论理论的多容错编码故障分析算法，将平

* Supported by the National Science Foundation of China under Grant No.90612001(国家自然科学基金); the Science Foundation of Tianjin under Grant No.043185111-14 (天津市科技发展计划重点项目)

作者简介: 王刚(1974－),男,博士,副教授,主要研究领域为存储系统,并行计算; 葛广贺(1979－),男,硕士生,主要研究领域为存储系统可靠性;邓万禧(1982－),男,硕士生,主要研究领域为存储系统可靠性。刘晓光(1974－),男,博士后,副教授,主要研究领域为存储系统,智能计算; 刘璟(1942－),男,硕士,教授,博士生导师,主要研究领域为算法设计与分析, 存储系统,并行与分布式系统.

凡算法指数阶的时间复杂性降低为多项式阶，大大提高了仿真模型的效率。我们实现了仿真模型和故障分析算法，仿真实验结果表明，新方法确实有效提高了多容错编码可靠性分析的准确性和效率。

关键词: 多容错编码;可靠性;Markov 模型;顺序蒙特卡罗模型;图表示法

中图法分类号: TP302 文献标识码: A

1 引言

数据存储一直是计算机科学研究的重要内容，特别是磁盘阵列技术（Redundant Arrays of Inexpensive Disks，RAID）[1]的提出，通过并行化方法、数据冗余机制，大大缓解了单一硬盘在容量、性能和可靠性上的局限，为大容量数据的存储提出了解决方案。磁盘阵列技术的相关研究中，容错编码是一个非常基础、非常重要的领域。近年来，网络技术的加入，使存储应用的形式越来越多样化，但大多数存储系统仍旧利用容错编码技术提供高可靠性和高性能。目前，存储系统呈现出网络化、分布式、超大规模等趋势，而硬盘技术则表现出容量提高迅速，性能和可靠性提高缓慢的趋势。这些影响系统可靠性的负面因素，使 RAID5 等单容错编码已经难以满足新型大规模存储应用在可靠性、可用性上的需求，推动了多容错编码的研究。

本文设计了一种利用顺序蒙特卡罗仿真模型分析存储系统多容错编码可靠性的方法，与传统的求解 Markov 模型的方法相比，新方法不仅能对 MDS 码进行简单分析，对非 MDS 码的复杂容错特性、多种故障因素和不同故障概率分布均能很好地进行仿真。本文还提出了基于图论理论的多容错编码故障分析算法，将平凡算法指数阶的时间复杂性降低为多项式阶，大大提高了仿真模型的效率。

本文内容组织如下，第二节我们会概述多容错编码的相关工作，重点分析 MDS 码和非 MDS 码的划分及其对可靠性分析的影响，还讨论了可靠性评价指标和分析方法。第三节我们主要介绍顺序蒙特卡罗模型和我们基于图论的多容错编码故障分析算法。第四节是仿真实验结果及其分析分析，主要对蒙特卡罗模型的正确性进行了验证，分析了它相对于求解 Markov 模型方法的优点，并利用它对多种容错编码方案的可靠性进行了对比分析。最后一节我们对所作研究工作进行了总结，并展望了今后的工作方向。

2 相关研究

存储系统多容错编码的研究自上世纪 90 年代初就已出现，近 20 年来出现了大量研究成果。Reed-Solomon 编码[2]是较早出现的一种多容错编码，双容错 RS 码被用于 RAID6 结构中，得到了较为广泛的应用。RS 码的校验磁盘开销和更新代价都是最优的，而且不局限于双容错，对于任意的 t 容错（t>2）需求，RS 码都有一致的解决方案。RS 码的缺点是，其编码/解码运算是有限域上的运算，须使用专用硬件，才能获得较好的编码/解码性能。

为解决 RS 码的缺陷，Gibson 等人提出了多维编码、full-2/full-3 码、steiner 码、addtivie-3 等一系列二进制线性码（binary linear codes）[3]。这类编码的校验运算只使用奇偶校验，通过将数据磁盘（单元）划分为若干重叠的校验组（每个校验组相当于一个 RAID5 条纹，每个数据（校验）单元参与多个校验组）来获得多容错能力。线性码的最大问题是冗余率较差，对于中小规模的存储系统尤为突出。

EVENODD[4]是第一个只使用奇偶校验，同时还能达到最优冗余的双容错编码方案。它将数据单元排列为方阵，在水平方向和对角线方向组织校验组，计算出一列水平校验和一列对角线校验（每列数据/校验单元放置在一个磁盘上），因此这类编码也被称为阵列码（array codes）。Blaum 等人还将 EVENODD 推广到多容错情况，他们的工作既是开创性的，也可以说是本领域最重要的。之后很多多容错编码的研究成果，都可看作 EVENODD 及其推广的变形。如双容错阵列码 X-Code[5]，选取主、从两个对角线方向组织校验

组，它是一种垂直码（vertical codes，每个磁盘混合存放数据单元和校验单元，独立存放方式，如 EVENODD，称为水平码，horizontal codes）。如 STAR-Code [6]，三容错水平码，在水平、主从两个对角线这三个方向上组织校验组。RDP 码[7]也是一种双容错水平码，它的校验方向与 EVENODD 相同，差异在于它是校验相关的（有校验单元还参与其他校验组），而 EVENODD 是校验无关的。类似成果还有很多，不再一一赘述。

传统的阵列码研究追求最优冗余率，但最近的研究表明这可能会导致较差的分布式性能。Hafner 提出的 WEAVER Code[8]，冗余率最好也只能达到 50%，但具有很小的校验组规模和极好的局部性，非常适合于分布式存储应用。已经得到的 WEAVER Code 编码方案最大可达到 12 容错能力，但其构造区别于其他编码的确定性公式化构造，需大量运算搜索可行编码个体。

实际上，对于可靠性研究来说，已有的多容错编码可分为两类。一类是以 RS 码为代表的 MDS 码（Maximum Distance Separable），即最小海明距离达到 Singleton 界，其冗余率是最优的，t 个校验磁盘即可达到 t 容错能力。EVENODD 等 MDS（近 MDS）阵列码也属此类，无论是容错特性还是系统可靠性，这些编码与 RS 码都没有任何差异。因此下文进行可靠性分析时，此类编码均以 RS 码（RAID6）作为代表。另一类是简单线性码、非最优冗余阵列码等非 MDS 码，这些编码冗余率差，似乎无法与 MDS 码抗衡。但近来，越来越多的研究表明，它们在可靠性、性能上的一些特殊优势有助于解决大规模分布式存储系统所面临的可靠性、性能上的一些问题。此类编码与 MDS 码的一个重要区别是，一个 t 容错的非 MDS 码，其容错能力往往不局限于 t 故障恢复，对于很多 k（k>t）故障，也可恢复。非 MDS 码的这种宽范围容错特性是相当复杂的，而且对于不同非 MDS 码其容错特性的特点也是不一样的，因此用传统的求解 Markov 模型的方式分析这类编码的可靠性，是非常困难的[9]。下文可靠性分析的重点，也是针对非 MDS 码的复杂容错特性。

容错编码的描述方法是非常基础的一个问题，恰当的描述方法对容错编码设计、性能和可靠性分析与优化的研究，都能起到很好的辅助作用。RS 码以有限域上运算的形式描述，编码性质可通过有限域理论加以研究。研究者关注更多的是线性码和阵列码的描述方法。Gibson 等人在文献[3]中详细论述了线性码的校验矩阵表示法（parity check matrix），对于 n 个数据磁盘，m 个校验磁盘的线性码，可用一个 m×(n+m)的 0/1 矩阵进行描述。每行代表一个校验组（校验磁盘），前 n 列代表 n 个数据磁盘，后 m 列表示 m 个校验磁盘。矩阵元素为 1 表示相应列的磁盘参与相应行的校验组，为 0 表示不参与。线性码的一些性质，即可用校验矩阵的性质进行表示：如，每行的重量即为对应校验组规模，每列重量即为对应数据磁盘参与的校验组数目，二维码、full2 码之类校验无关线性码的后 m 列形成一个单位阵等等。最重要的，某个 k 故障可否恢复，等价于 k 个故障磁盘对应的 k 列是否线性无关——等价于这 k 列是否没有任何非空子集之和（GF[2]上加法运算，即 XOR 运算）为 0 向量。这为多容错线性码的容错特性的研究提供了坚实的数学基础。对于阵列码，显然上述方法和结论稍加改动，仍然适用。

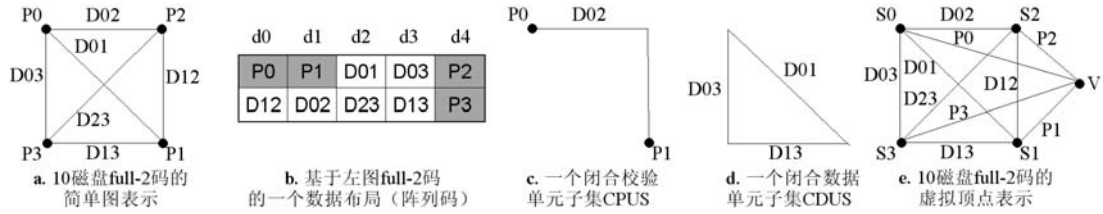


图 1 简单图描述法

另一种重要的描述方法是图描述法，Gibson 等人早在文献[3]中就提出用简单图表示 full-2 码，随后一些国内外文献也提到了类似的思路。南开大学并行与分布式系统软件实验室对容错编码图表示法及借助图论理论研究编码性质和多容错阵列码设计等问题，进行了系统的研究。对于校验无关的双容错线性码，可用简单图完美描述：用顶点表示校验磁盘（校验单元，校验组），用边表示数据磁盘，边与顶点的邻接关系

表示数据磁盘与校验组的参与关系[10]。图 1a 给出了 10 磁盘 full-2 码的简单图描述。容易看出，阵列码中条纹单元校验分组关系同样可用这样的简单图进行描述，由此也可知道，每个阵列码实际都是基于某个线性码的数据布局。简单图无法表示阵列码中条纹单元的磁盘分布，但很明显，阵列码的磁盘布局与图的划分是一一对应的。阵列码是否具有需要的容错能力，可借助图的分解的性质加以研究[10]：

定理 1. 一个阵列码（数据布局）具有双容错能力，当且仅当，其对应的图划分中，任何两个子图的并均不包含如下两种类型的子图：

1) 一条路（一个相邻边序列）及其两个端点，我们称这种不可恢复故障为闭合校验单元子集（Closed Parity Units Subset，CPUS）。

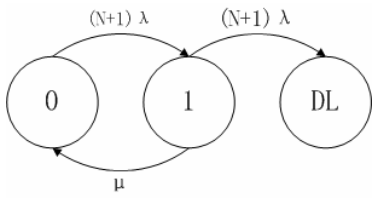
2) 一个圈，我们称之为闭合数据单元子集（Closed Data Units Subset，CDUS）。

图 1c 和图 1d 给出了 CPUS 和 CDUS 的例子，它们分别对应图 1b 阵列码磁盘 0 和磁盘 1 故障及磁盘 2 和磁盘 3 故障。显然，在一个磁盘保存一个数据单元，并不意味着所属的校验单元也保存在此磁盘。因此对相应的图划分，一个子图包含一个边，并不意味着包含其端点，这与图论的一般表述方法有些差异。文献[11]中提出了一种虚拟顶点表示法：增加一个虚拟顶点，其他顶点只表示校验组，校验单元改用边表示——所属校验组和虚拟顶点间的边。图 1e 给出了 10 磁盘 full-2 码的虚拟顶点表示法，能有效解决此问题，CPUS 和 CDUS 合二为一，均对应圈。显然，定理 1 也完全可以用来判断简单线性码的故障是否可恢复，本文第 3 节提出了基于定理 1 的高效的不可恢复故障检测算法。

以往的研究中，对容错编码的可靠性主要有两种评价指标。一是编码的“容错能力”，如，RAID5 是“单容错”的，RAID6 是“双容错”的等等。即，编码可以保证恢复多少个磁盘故障。这一指标在编码设计中就已确定，因此很容易获得。容错能力指标可以很清晰地描述出 MDS 码的故障恢复能力，但对于非 MDS 码，就远远不够了。传统的容错能力指标实际上是一种“最低容错能力”，或者称“海明容错能力”（hamming fault tolerance）[9]，“门限容错能力”，即容错编码可保证 100%恢复的故障数的最大值。如前所述，t 容错的非 MDS 码，除了可以保证恢复所有 t 故障外，还可恢复很多 t+1 故障、t+2 故障、…。虽然不能保证 k（k>t）故障可 100%恢复，但这种故障恢复能力对可靠性的作用显而易见。近年来，研究者尝试将广泛用于通信领域的 LPDC 码用于存储系统。LDPC 码的研究中采用一种称为开销因子（overhead factor）的评价标准，对于存储系统，开销因子可以转换为“平均容错能力”（average fault tolerance）[12]，这一指标可以比较公平地描述非 MDS 码的故障恢复能力。我们认为，可以仿照算法时间复杂性分析中“最好、最坏、平均时间复杂性”的评价标准，对容错编码设定“最低、最高、平均容错能力”评价指标，可全面评价编码故障恢复能力。

容错能力评价的是编码的对磁盘故障的恢复能力高低，从系统可靠性角度，可以用平均数据丢失时间（Mean Time To Data Loss，MTTDL）来评价。传统的分析方法是为存储系统故障发生和恢复建立 Markov 模型，通过对系统崩溃概率的分析，来获取 MTTDL。以往研究认为，磁盘的故障事件是相互独立的随机变量序列，且服从均值为 1/λ 的指数分布[13]，其中 λ=1/MTTF（Mean Time To Failure，磁盘的平均故障时间）。根据 Poisson 过程的定义有：P(“death” before t) = 1- $e^{-\lambda t}$=1 - $e^{-t/M}$ 。对于一个负指数函数，类似的可以进行近似化简：$e^{-t/M}$ ≈ 1- t/M。则可以得到：Prob(“death” before t) = t/M。类似的，研究者认为磁盘阵列系统的故障事件同样是相互独立的随机变量序列，且服从均值为 1/λ’的指数分布，其中 λ’=1/MTTDL。

因此，可为存储系统的故障发生和故障修复建立 Markov 模型，通过求解系统故障状态的发生概率，来计算 MTTDL 的数学期望。图 2 给出了一个单一校验条纹的 RAID5 阵列的 Markov 模型。



状态 0：无故障状态　　状态 1：发生一个磁盘故障　　状态 DL：数据丢失　　N+1：磁盘数

$MTTF_{disk}$：磁盘平均无故障时间　　故障率 λ=1/ $MTTF_{disk}$

$MTTR_{disk}$：磁盘平均修复时间　　修复率 μ=1/ $MTTR_{disk}$

图 2 单条纹 RAID5 的 Markov 模型

求解 DL 稳态概率，取倒数即可得到系统 MTTDL 的数学期望：

$$MTTDL=\frac{(2N+1)\lambda+\mu}{N(N+1)\lambda^2} \quad \text{（公式 1）}$$

由于 λ 远小于 μ，可近似为：

$$MTTDL=\frac{\mu}{N(N+1)\lambda^2}=\frac{MTTF_{disk}{}^2}{N(N+1)MTTR_{disk}} \quad \text{（公式 2）}$$

类似可得 RAID6 编码的 MTTDL：

$$MTTDL=\frac{(2N+2)(N+1)\lambda^2+(\mu+N\lambda)(N+2)\lambda+N\mu\lambda+\mu^2}{N(N+1)(N+2)\lambda^3} \quad \text{（公式 3）}$$

可近似为：

$$MTTDL=\frac{MTTF_{disk}{}^3}{N(N+1)(N+2)MTTR_{disk}{}^2} \quad \text{（公式 4）}$$

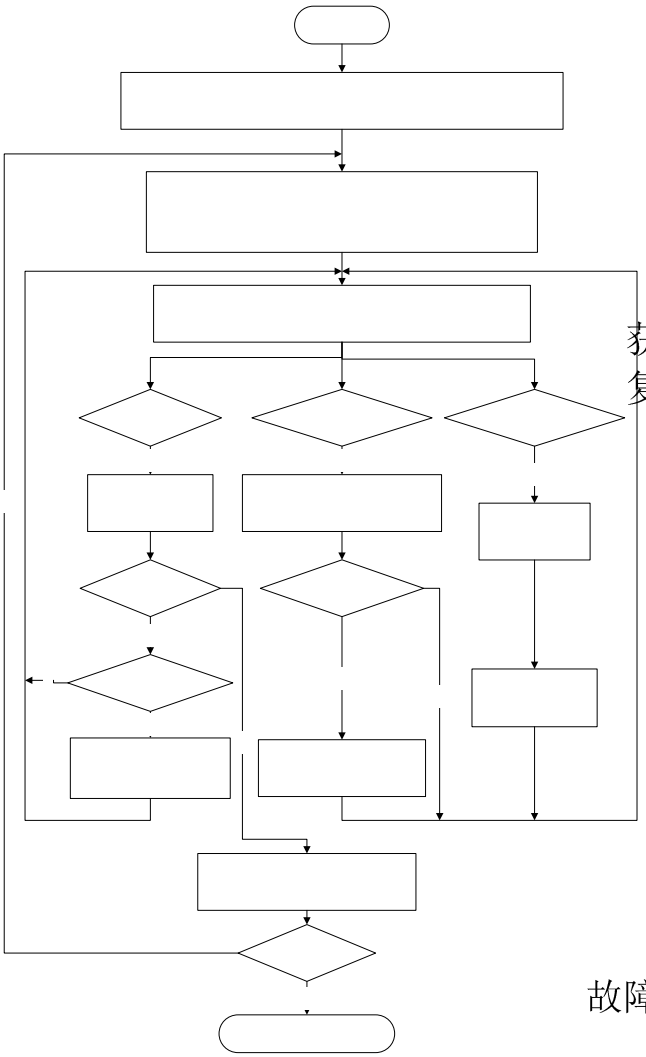


图 3 顺序蒙特卡罗仿真模型算法

一般可靠性的研究即利用公式 2 和公式 4 分析单容错和双容错磁盘阵列的可靠性。但这种方法有着非常大的局限性，对于非 MDS 码的复杂容错特性，对于故障相关性等复杂故障发生模式，对任意不可恢复读写故障等复杂故障类型，以及非指数的故障事件分布，建立 Markov 就已非常复杂，求出如公式 1－公式 4 这样的一般性的解析公式几乎是不可能的。例如，Hafner 等人利用 Markov 方法对 WEAVER Code 进行可靠性分析，若编码的最低容错能力为 t，最大容错能力为 T，则须建立 T+2 个状态，且须对所有 k（t<k<=T），穷举可恢复 k 故障与全部 k 故障的比例[9]。根本无法求解通用的 MTTDL 公式，只能对编码个例，利用 MATLAB 等工具求解，但也无法保证肯定可解。更为麻烦的是，对于不同编码个例（如规模发生变化），上述过程须重新来过一遍，计算量是极其巨大的。因此，我们考虑采用顺序蒙特卡罗模型仿真方法，其优点是，无论是复杂的编码特性，还是复杂的故障模型，均可顺畅地加入，对求解的难度几乎没有影响。

3 多容错编码可靠性仿真模型

3.1 顺序蒙特卡罗仿真模型

蒙特卡罗方法的基本思想是：当所求解问题是某种随机事件出现的概率，或者是某个随机变量的期望

值时，通过实验的方法，以这种事件出现的频率估计这一随机事件的概率，或者得到这个随机变量的某些数字特征，并将其作为问题的解。基于蒙特卡罗方法，我们设计了存储系统可靠性仿真分析模型。

在顺序蒙特卡罗模型中，系统事件按时间序进行模拟。为模拟存储系统故障发生和恢复，我们对磁盘的故障事件、故障修复完毕事件进行采样，如系统采用人工检查模式，还需采样检查事件，并对采样得到的事件按时间序进行处理。如前所述，磁盘的故障事件是相互独立的随机变量序列，且服从均值为 1/λ 的指数分布。因此，磁盘故障采样服从均值为 1/λ 的指数分布。由于故障修复时间和人工检查间隔时间均取定值，因此事件采样也取定值。磁盘故障事件会触发系统崩溃检测，也可能触发故障修复完毕事件（即时修复模式）。人工检查事件会触发故障修复完毕事件（若有故障发生）。而故障修复完毕事件会触发新磁盘故障事件的采样。当系统崩溃时，当前时间作为此次仿真的 MTTDL 结果。重复仿真多次，对结果进行统计分析，得到 MTTDL 的数学期望等所需结果。算法描述如下：

算法 1(顺序蒙特卡罗模型仿真算法).

输入:编码方案 C、系统磁盘数 N、磁盘平均无故障时间 MTTF、平均修复时间 MTTR、仿真次数 R、故障检查恢复策略、人工检查周期 P。

输出:系统平均数据丢失时间 MTTDL 的数学期望。

方法:流程图如图 3。

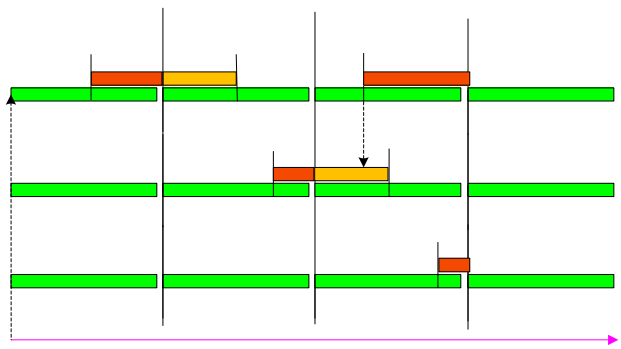


图 4 顺序蒙特卡罗模型仿真实例

图 4 给出了利用顺序蒙特卡罗模型仿真一个 3 磁盘的 RAID5 系统的可靠性的一次可能的运行实例。仿真开始时，为每个磁盘采样一个故障事件。首先处理最早发生的故障 1，发现系统未崩溃，由于采取人工定期检查恢复模式，不触发故障修复。随后处理第一个人工检查事件，发现磁盘 1 处于故障状态，启动修复，采样一个故障修复完毕事件。该事件早于其他事件，因此下来对它进行处理，为替换后的新磁盘采样一个故障事件——故障 4。接下来应处理的事件是故障 2 和检查 2，与故障 1 和检查 1 的处理类似。随后处理故障 4 时，发现当前故障磁盘数为 2，系统崩溃，仿真结束，故障 4 发生时间为此次仿真得到的 MTTDL。

顺序蒙特卡罗模型相对于 Markov 模型求解方法有很多优点：1）可以方便地增加对各种编码方案的支持，MDS 码和非 MDS 码仿真难度相差不大，求解 Markov 的方法对新的编码则要重新进行所有工作，对非 MDS 码分析难度极大；2）可方便地支持及时修复、定期检查等不同的修复模式，求解 Markov 模型方法对定期检查方式的分析非常困难；3）对于故障相关性等故障模式的分析，仿真方法较之 Markov 模型求解方法要简单很多；4）最近有研究者提出，磁盘和系统故障不是服从指数分布的随机变量，故障率是时间相关的[14]，Markov 模型方法根本无法处理这种情况，但仿真方法通过简单的算法修改即可支持这种故障

检查点1 检

故障1 修复

磁盘1

分布的改变。

3.2 基于图描述法的系统崩溃检测

在顺序蒙特卡罗方法中，系统崩溃检测是非常重要的一部分。对于 RAID5、RAID6 等 MDS 码，新磁盘故障是否导致不可恢复故障，是很容易判断的，检查故障数是否大于编码的最低容错能力（也是平均、最高）即可。

对于非 MDS 码，不可恢复故障的检测是很困难的。平凡算法是利用文献[3]中的校验矩阵表示法，检测故障磁盘（列）的线性无关性。对于故障磁盘集合，穷举其所有非空子集，计算列向量之和，若有为 0 者，此故障为不可恢复故障，否则，若所有非空列向量子集和均非 0，则此故障可恢复。假定编码数据单元数为 n，校验单元数为 m，故障磁盘数为 k，最低容错能力为 t，考虑到规模<=t 的子集无需考虑，则算法需计算 $2^k-1-\sum_{i=1}^{t}C_k^i$ 个子集之和，时间复杂性是指数阶的 O(m*2^k)。若编码最高容错能力为 T，显然 k 的取值范围在 1<=k<=T+1。对于二维码、full-2 码等非 MDS 码，T 可能与 n 和 m 在同一数量级，算法时间复杂性可能达到磁盘数的指数阶。若存储系统规模较大，运行时间是难以接受的，大大削弱了顺序蒙特卡罗模型的优势。

基于定理 1，我们可以设计出一个针对双容错编码的高效的不可恢复故障判定算法。对一个 k 故障，使用虚拟顶点表示法，构造它对应的子图。检测子图是否包含圈，若包含，则此 k 故障无法恢复，否则，属可恢复故障。而一个图是否包含圈的检测，是相当简单的，利用深度优先搜索或宽度优先搜索即可。我们通过一个与邻接列表作用类似的列表数组 failed_stripes[]维护故障磁盘对应的子图，当磁盘故障事件发生，获取故障磁盘对应的两个校验组 s 和 t，将 s 添加至 failed_stripes[t]，将 t 添加至 failed_stripes[s]，当故障修复完毕事件发生，对列表进行删除操作。故障检测算法如下伪代码所示：

算法 2(双容错编码不可恢复故障检测算法).

输入:故障磁盘集合对应子图 G=(V, E)（虚拟顶点表示法），最新故障磁盘编号 d。

输出:Yes——不可恢复故障；No——可恢复故障。

方法:

1) V_visit[]←false, E_visit[]←false
2) 获取 d 所属校验组编号 s、t
3) V_visit[s], V_visit[t]←true, E_visit[(s, t)]←false
4) DFS(s)
   对 failed_stripes[s]中所有顶点 s’，执行 a)—b)
   a) 若 E_visit[(s, s’)]=true 继续循环
   b) 若 V_visit[s’]=true，算法终止，返回 Yes
   c) V_visit[s’]←true, E_visit[(s, s’)]←true
   d) 递归调用 DFS(s’)

显然，算法的时间复杂性为 O(|E|+|V|)，即，与磁盘数呈线性关系，极大改进了平凡算法的性能。

4 仿真实验结果与分析

4.1 顺序蒙特卡罗的实现与仿真实验设定

我们在 Linux 平台实现了顺序蒙特卡罗模型。程序主框架为算法 1 所示的事件驱动模式，实现了算法 2 的不可恢复故障检测算法，对二维码、full-2 码进行崩溃判定，对 RAID5、RAID6 等 MDS 码也实现了相应的检测算法。故障事件采样是另一个非常重要的模块，采样结果如能很好吻合指数分布，会使仿真结果更为精确。我们采用在开源社区广泛使用的 GSL（GNU Science Library）[15]来生成服从指数分布的随机数序列，仿真结果的统计分析也使用 GSL 来实现。

我们利用仿真程序对几种存储系统多容错编码的可靠性进行了仿真分析，测试了不同编码、磁盘数、MTTF、MTTR 等因素对可靠性的影响。测试了 RAID5、RAID6、二维码、full-2 码等编码方案，其中二维码的数据磁盘均设定为方阵，磁盘数测试了 10—200 的范围，MTTF 设定为 50000 小时[15]，MTTR 设为 12 和 18 小时，为提高结果的准确性，每个仿真重复 10000 次，对结果进行统计分析，计算 MTTDL 的数学期望。

4.2 容错能力分析

在进行仿真实验之前，我们首先对几种编码方案的容错能力进行一下分析，会对它们的可靠性高低有一个直观认识。RAID5、RAID6 两种 MDS 码很简单，前者的最低、最高、平均容错能力均为 1，后者均为 2。二维码和 full-2 码的最低容错能力均为 2，最高容错能力可由定理 1 获得。

对一个 n 个校验组的 full-2 码，采用虚拟顶点表示法，可表示为一个完全图 K_{n+1}。由定理 1，若子图边数量>=n+1，必然会形成圈。也就是说，若故障磁盘数>=n+1，必然包含不可恢复故障。而我们确实可以构造出可恢复的 n 故障，比如，以虚拟顶点为一个端点的 K_{n+1} 长度为 n 的一条路，对应 n-1 个数据磁盘和一个校验磁盘。因此，full-2 码的最高容错能力为 n。

对于一个有 n 个水平校验组，n 个垂直校验组的二维码，用简单图表示法，可描述为一个完全二部图 $K_{n,n}$。类似的，若子图边的数量和顶点数量之和>=2n+1，由定理 1，必然构成 CPUS 或 CDUS。而将 $K_{n,n}$ 的某个哈密尔顿圈删除一条边，添加该边的某个端点，由定理 1，这个子图对应的 2n 故障是可恢复的。因此，二维码的最高容错能力为 2n。表 1 给出了这几种编码的容错能力对比。

表 1. 几种编码方案容错能力对比

编码方案	磁盘数	最低容错能力	最高容错能力
RAID5	N	1	1
RAID6	N	2	2
二维码	$N=n^2+2n$	2	$2n=O(\sqrt{N})$
full-码	N=n*(n+1)/2	2	$n=O(\sqrt{N})$

二维码和 full-2 码的平均容错能力的计算是非常复杂的，很难得到一般性的解析公式。可将文献[12]中的开销因子计算算法转换为计算平均容错能力的算法，但算法时间复杂性是指数阶的，且需对每个规模执行算法计算特定系统的平均容错能力，计算量是非常巨大的。

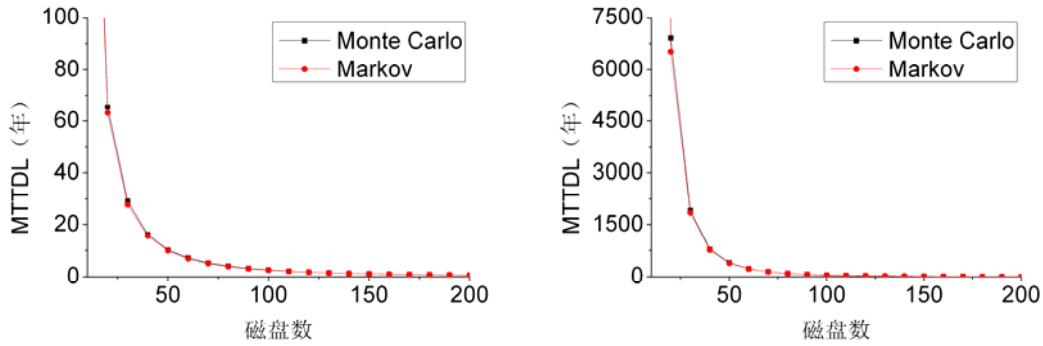
4.3 模型正确性验证

为验证蒙特卡罗模型的正确性，我们修改了算法 1 的故障修复模式，使之与 Markov 模型的故障修复模式完全吻合。利用修改算法进行仿真实验，将仿真结果与 Markov 模型求解结果进行比较，以验证仿真模型的正确性。注意到，Markov 模型的故障修复模式并非简单的及时修复模式，故障修复并非以磁盘为对象独立进行的，而是以系统为对象进行整体修复，任何时刻只有一个修复动作在进行。因此，将算法 1 修改为：当磁盘故障事件发生时，若有正在进行的修复动作，则将故障修复完毕事件从事件队列删除，为新故障磁盘加入发生于 curr_time+MTTR 的故障修复完毕事件；当故障修复完毕事件发生，若为逐步恢复模式[9]，且尚有故障磁盘，为其加入发生于 curr_time+MTTR 的故障修复完毕事件，若为一次性恢复模式[9]，

则系统恢复到正常状态。图 5a 给出了对 20－200 个磁盘的 RAID5 阵列，仿真结果与公式 1 计算结果的对比，MTTF 取 50000 小时，MTTR 取 12 小时，图 5b 给出的时 RAID6 阵列的对比结果，MTTR 取 18 小时。非常明显，两种方法的结果曲线几乎吻合，结果最大差距不超过 5%，可以认为是实验误差。当 MTTF、MTTR 等参数变化时，有类似的结果，限于篇幅，这里不一一列出。可见，仿真模型的正确性是有保证的。

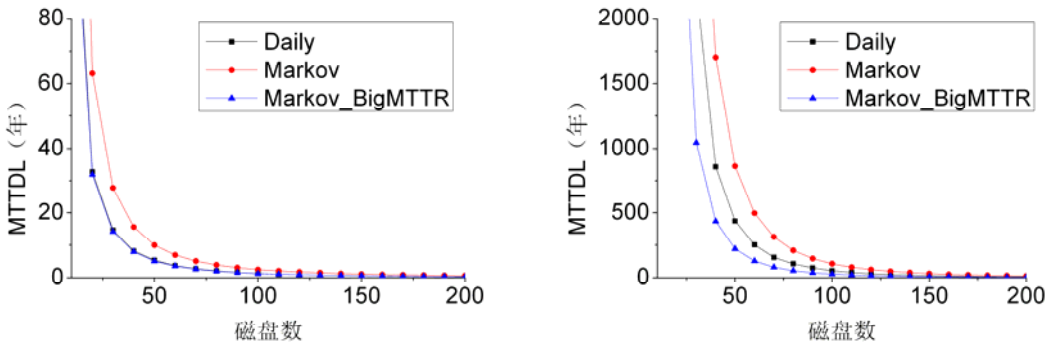
4.4 人工定期检查修复模式的仿真

很多大型存储系统，由于结构的复杂性，以及出于成本的考虑，不采用配备热闲置盘，故障自动即时恢复的方式。而是采用人工定期检查，手工替换故障盘，启动修复的模式。由算法 1 容易看出，这种模式在顺序蒙特卡罗仿真模型中很容易实现，但用 Markov 模型描述是非常困难的。图 6 给出了对于 RAID5 和 RAID6 系统，利用仿真方法得到的人工检查模式结果（Daily 曲线），与用公式 1 和公式 3 计算结果的对比（Markov 曲线），MTTF 取 50000 小时和 100000 小时，MTTR 取 12 小时，人工检查周期为 24 小时。可以看到，两种方法的结果一直在 50%左右。对于检查周期为 P 的情况，是否可认为故障修复延迟的数学期望为 P/2，令 MTTR+P/2 为真正的修复时间，由此利用 Markov 模型是否可以求得定期检查的准确结果呢？我们对此进行了计算，图 6 中 Markov_BigMTTR 曲线即为计算结果。RAID5 的结果似乎验证了这种想法，但是看一下图 6b，非常明显，对于 RAID6 编码，即便调整 MTTR，求解 Markov 模型得到结果也与仿真结果有巨大差异，两者差距在两倍左右。究其原因，对 RAID5 编码，只有单故障的修复，上述想法是合理的。但对多容错编码，定期检测修复有单故障修复和多故障修复之分，恢复故障数目不同，上述思路就会有很大差异，利用 Markov 模型对此建模，几乎是不可能的。



a. RAID5　　b. RAID6

图 5 顺序蒙特卡罗模型正确性验证



a. RAID5　　b. RAID6

图 6 人工定期检查模式

4.5 编码可靠性对比

图 7 和图 8 给出了各种编码的可靠性对比，容易看出，随磁盘数的增加，RAID5 和 RAID6 的可靠性急剧下降（大致与磁盘数呈平方和立方关系下降）。RAID5 在磁盘数为 20 时，10 年故障率就已达到 26%，磁盘数大于 30，已超过 50%，当磁盘数大于 80 时，已接近 100%。RAID6 虽然明显优于 RAID5，但在磁盘数大于 100 时，10 年故障率也已超过 16%。而我们尚未考虑不可恢复读写故障、磁盘故障相关性等因素，故障恢复时间也取的是较为理想的值，这表明，RAID5 和 RAID6 难以满足大规模存储系统的可靠性需求。而 full-2 码和二维码在磁盘数接近 100 时，10 年故障率仅仅分别为 0.1%和 0.02%，与 RAID6 相差几个数量级。更为重要的是，full-2 码和二维码在磁盘数增加的情况下，可靠性下降较为平缓。这种特性，可保证即使存储系统不断扩充规模，也能提供满意的可靠性。因此，虽然 full-2 码和二维码冗余率较差，但考虑到这一缺点随磁盘数增加会有所缓解，而且目前硬盘价格下降非常迅速，对于大规模存储系统，这两种编码应该是比 RAID5、RAID6 这类 MDS 码更好的选择。

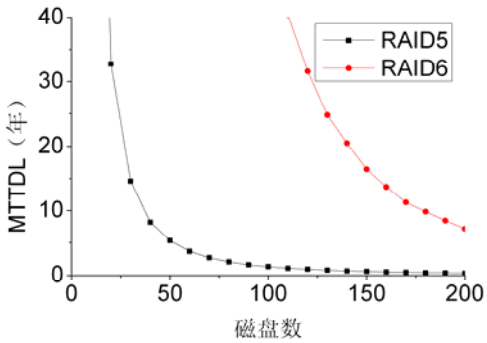


图 7. RAID5 和 RAID6 可靠性结果

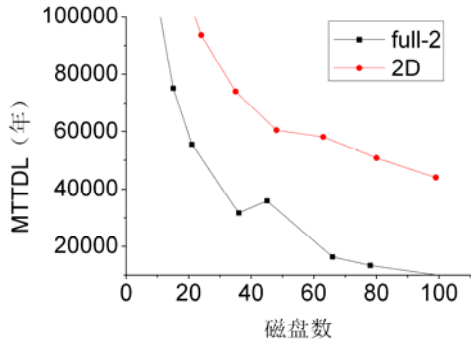


图 8. 二维码和 full-2 码可靠性结果

4.6 full-2码和二维码不可恢复故障类型分析

如第二节所述，双容错线性码的故障有两类：CPUS 和 CDUS。我们对 full-2 码和二维码进行了实验，分析了两类故障出现的概率。MTTF 设定为 10000 小时，MTTR 设定为 12 小时，仿真次数 1000 次。图 9 给出了实验结果，容易看出，二维码绝大部分是由于 CPUS 导致系统崩溃，而 full-2 码的崩溃则大多由 CDUS 所导致。

造成这种现象的原因是，我们考虑简单图表示法，一个 2n 个校验组的二维码对应完全二部图 $K_{n,n}$。二部图中，能够形成的 CDUS（圈）的长度只可能为偶数，范围在 4～2n 之间。而长度为奇数、偶数的 CPUS 均有可能（这里长度指边——数据单元的数量），范围在 2～2n-1 之间。在 CDUS 和 CPUS 之间存在着一对多的映射关系：对一个长度为 2k 的 CDUS，删除任一条边，添加这条边的两个端点，即得到一个长度 2k-1 的 CPUS，而不同 CDUS 删除不同边，得到的 CPUS 都是唯一的，因此一个长度 2k 的 CDUS 对应 2k 个不同的长度为 2k-1 的 CPUS；若删除任意两条相邻边，加入剩余路的两个端点，则得到长度为 2k-2 的 CPUS，但一个这样的 CPUS 对应 n-k+1 个不同的 CDUS，即一个长度 2k 的 CDUS 对应 2k 个不同的长度为 2k-2 的 CPUS，还应除以因子 n-k+1。因此，我们有如下结论。

定理 2. 一个有 n 个水平校验组，n 个垂直校验组的二维码，其 CDUS 的数量与 CPUS 的数量的比例为.

$$\sum_{k=2}^{n}(C_n^k \times C_n^k \times \frac{P_{k-1} \times P_k}{2}) / (\sum_{k=2}^{n}(C_n^k \times C_n^k \times \frac{P_{k-1} \times P_k}{2}) \times (2k + \frac{2k}{n-k+1}) + n^2) \qquad \text{（公式 5）}$$

证明：二维码对应完全二部图 $K_{n,n}=(V, W, E)$。

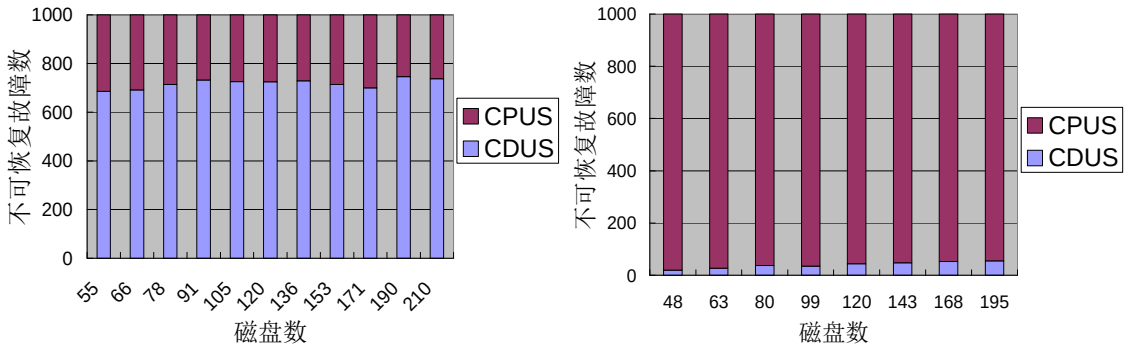
长度为 2k 的 CDUS 必然是有 k 个顶点来自 V，k 个顶点来自 W，顶点组合数量为 $C_n^k \times C_n^k$。

对于选定的一组 2k 个顶点，CDUS 的数量与顶点的全排列数量相关。由于是构造圈，因此只需考虑 2k-1 顶点的排列。对于二部图，路径顶点序列在 V、W 间穿梭，因此数量为 $P_{k-1} \times P_k$。再去掉顺、逆时针对称的情况，除以 2 即可。由此得到 CDUS 数量即为公式 5 的分子部分。

而如前所述，CDUS 与 CPUS 间存在一对多关系，另有 n^2 个长度为 1 的 CPUS 无 CDUS 与之对应，因此得到 CPUS 数量为分母部分。

□

公式 5 的计算结果与我们的仿真结果是比较接近的，限于篇幅，这里不再列出。但两者有一个比较大的差别，当 n>2，公式 5 是递减的，而图 9b 的仿真结果明显是递增的。我们认为，这种现象的原因是，一个随机的 k 故障形成一个长度（接近）k 的 CDUS 的概率显然远远小于长度（接近）k 的 CPUS，但一个 k 故障完全可能形成较小规模的 CDUS 或 CPUS。而磁盘故障是相继发生的，当一个可恢复 k 故障演变成一个 k+1 故障时，实际上多数情况下演变为可恢复 k+1 故障的概率远远高于演变为 CDUS 和 CPUS。当数据磁盘数大大超过校验磁盘时，这种演变会使故障磁盘中数据盘的比例远远大于校验盘，从而导致 CDUS 出现的概率增加。较小规模的二维码，校验盘数与数据盘数是相当的，当规模增大，数据盘数逐渐远远超过校验盘数，从而造成实际 CDUS 发生概率与公式 5 呈相反趋势。



a. full-2 码　　b. 二维码

图 9 不可恢复故障类型

再看 full-2 码，如果使用虚拟顶点表示法的话，n 个校验组的 full-2 码对应完全图 K_{n+1}。对于 CDUS 和 CPUS 比例，我们有如下结论。

定理 3. 一个有 n 个校验组的 full-2 码，其 CDUS 的数量与 CPUS 的数量的比例为.

$$\sum_{k=3}^{n}(C_n^k \times \frac{P_{k-1}}{2}) / (C_n^2 \sum_{k=0}^{n-2} C_{n-2}^k P_k) \qquad \text{（公式 6）}$$

证明：因为采用虚拟顶点表示法，因此 K_{n+1} 中有 n 个校验组顶点，CDUS 来自这些顶点组成的圈，长度范围 3～n，因此易知公式 6 分子部分为 CDUS 数量。

CPUS 对应包含虚拟顶点的圈，首先选定圈中虚拟顶点的两个相邻顶点，数量为 C_n^2。剩余 n-2 个顶点

中任意 k 子集（0<=k<=n-2）均可能包含在圈中，对每个子集，k 个顶点的每种排列都对应一个唯一的圈，共 P_k 种可能。因此 CPUS 数量即为分母部分。

□

对于相近的磁盘数，公式 6 的结果比公式 5 大两倍左右，由于完全图更为稠密，这是显然的。但与图 9a 仿真结果相比，相差较远。我们认为，原因与二维码的分析类似。我们所测试的规模中，数据磁盘数都远远大于校验磁盘数，再考虑到故障相继发生的情景，使得 CDUS 的实际发生概率超过了 CPUS。

5 结论

本文设计了一种利用顺序蒙特卡罗仿真模型分析存储系统多容错编码可靠性的方法，与传统的求解 Markov 模型的方法相比，新方法不仅能对 MDS 码进行简单分析，对非 MDS 码的复杂容错特性、多故障因素和不同故障概率分布均能很好地进行仿真。本文还提出了基于图论理论的多容错编码故障分析算法，将平凡算法指数阶的时间复杂性降低为多项式阶，大大提高了仿真模型的效率。仿真结果表明，对于 Markov 模型能分析的实例，仿真方法都能正确分析，而对上述 Markov 模型难以处理的因素，仿真方法也能较为容易地进行处理。我们对几种多容错编码进行了仿真实验，得到了一些有意义的结论。

在今后的工作中，还需进一步完善仿真模型：如，增加对不可恢复读写故障（即，非磁盘整体故障，而是某个磁盘区域故障）及利用磁盘清洗等手段修复此类故障的仿真支持；如，增加对故障相关性的仿真，这对网络存储系统结构是非常有意义的；再如，对近来研究者认为磁盘和系统故障非指数分布这一新的研究动向，也可考虑引入到仿真模型中。总体来讲，在仿真模型中增加对这些因素的仿真难度并不大。对于高可靠性的非 MDS 码，目前的仿真程序运行速度还比较慢，下一步需对算法进行优化，并考虑并行化等方法。还应增加对更多编码方案的支持，并进行大量的更为全面的仿真实验，通过实验结果指导多容错编码可靠性优化和编码设计的研究，这应该是最为重要的。此外，借助图论理论对非 MDS 码的容错性质进行更为深入的分析也是很有意义的工作。

致谢 感谢南开大学科学计算所对本文工作的支持.

References:

[1] Patterson, D. A., Gibson. G., and Katz, R. H., A Case for Redundant Arrays of Inexpensive Disks (RAID), In Proc. ACM SIGMOD, pages 109-116, ACM, 1988.

[2] J. S. Plank, “A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems”, Software - Practice & Experience, 27(9), September, 1997, pp. 995-1012.

[3] Lisa Hellerstein, Garth A. Gibson, Richard M. Karp, Randy H. Katz and David A. Patterson, “Coding techniques for handling failures in large disk arrays”, Algorithmica 1994 12(2/3): 182–208.

[4] M. Blaum, J. Brady, J. Bruck, J. Menon, EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures, IEEE Trans. Comput., 1995, 44(2): 192-202.

[5] L. Xu and J. Bruck, “X-Code: MDS Array Codes with Optimal Encoding,” IEEE Trans. on Information Theory, 45(1), 272-276, Jan, 1999.

[6] Cheng Huang, Lihao Xu, “STAR: An Efficient Coding Scheme for Correcting Triple Storage Node Failures”, 4th USENIX Conference on File and Storage Technologies San Francisco, 2005, pp. 197-210.

[7] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong and S. Sankar, “Row-Diagonal Parity for Double Disk Failure Correction”, Proc. of USENIX FAST 2004, Mar. 31 to Apr. 2, San Francisco,CA, USA.

[8] J. L. Hafner, “WEAVER Codes: Highly Fault Tolerant Erasure Codes for Storage Systems,” 4th Usenix Conference on File and Storage Technologies, December, 2005.

[9] J. L. Hafner, KK Rao, “Notes on Reliability Models for Non-MDS Erasure Codes”, IBM research report, RJ10391(A0610-035), October 24, 2006.

[10] Zhou Jie,Wang Gang, Liu Xiaoguang, Liu Jing, “The Study of Graph Decompositions and Placement of Parity and Data to Tolerate Two Failures in Disk Arrays: Conditions and Existance”, Chinese Journal of Computer, vol. 26, no. 10, pp. 1379-1386, Oct, 2003.

[11] Wang Gang, Dong Sha-sha, Liu Xiao-guang, Lin Sheng, Liu Jing, “Construct double-erasure-correcting Data Layout Using P1F”, ACTA ELECTRONICA SINICA, 2006, 34(12A), pp. 2447-2450.

[12] James S. Plank, Adam. L. Buchsbaum, Rebecca. L. Collins and Michael. G. Thomason, “Small Parity-Check Erasure Codes - Exploration and Observations,” DSN-05: International Conference on Dependable Systems and Networks, Yokohama, Japan, June, 2005.

[13] G. Gibson and D. Patterson, “Designing Disk Arrays for High Data Reliability,” Journal of Parallel and Distributed Computing, vol. 17, 1993, pp. 4-27.

[14] Jon G. Elerath, Michael Pecht, “Enhanced Reliability Modeling of RAID Storage Systems”, In Proc. 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07), pp. 175-184, IEEE 2007.

[15] Qin Xin, Ethan L. Miller, Thomas Schwarz, S.J, Darrell D. E. Long, “Reliability Mechanisms for Very Large Storage Systems”, Proceedings of the 20 th IEEE/11 th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS’03), 2003.

附中文参考文献:

[10] 周杰，王刚，刘晓光，刘璟，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，计算机学报，2003，26(10): 1379-1386。

[11] 王刚，董沙莎，刘晓光，刘璟，利用图的完全 1-因子分解构造双容错数据布局，电子学报，2006 年，第 34 卷第 12A 期，2447 页-2450 页。