并发维护下纠删码存储系统的持久性与可用性

Suayb S. Arslan

“一旦真理被发现，理解起来就很容易；关键在于发现它。”伽利略。版本 2.0 – 2023年1月

arXiv:2301.09057v1 [cs.PF] 22 Jan 2023

前言

本文档的初版写于 2014 年，旨在介绍可靠性理论的基础知识，并确定用于预测纠删码存储系统耐久性/可用性的理论工具类型。由于“系统”的定义过于宽泛，我们特别关注温存储和冷存储系统。在这些系统中，数据以分布式方式存储在不同的存储单元上，系统可以持续运行（全占空比），也可以不持续运行。

本文档的内容包括对基础知识的回顾、若干重要的改进型随机模型，以及我在该领域的几项相关研究成果。本文档的一项有趣贡献是引入了用于估算平均故障时间的最一般形式的马尔可夫模型。这项工作的部分成果后来发表于 *IEEE Transactionson Reliability*。本文还探讨了该一般模型闭式解的高精度近似。我们利用这类先进模型比较了不同策略下的各种存储配置。在后续章节中，我们还考虑了多维马尔可夫模型，以处理驱动器与介质分离的组合，例如光盘和磁带存储系统中的此类组合。不难预见，这种系统结构很可能成为未来 DNA存储库的一部分，因此具有大量有趣的应用。这项工作的部分成果发表于 *Elsevier Reliability and System Safety*。

文档末尾将介绍包含仿真建模的主题，以便进行更准确的估算，届时会指出简化的规范马尔可夫模型以及更复杂的马尔可夫模型的不足之处，这主要是由于马尔可夫性的平稳性和静态特征所致。在全文中，我们将重点讨论并发维护的系统，尽管对于一次只修复一个设备的系统，相关讨论仅需略作调整。本文档仍在编写中，未来版本可能会加入更新的模型和新颖的方法，以丰富现有内容。读者可能需要具备一些概率和编码理论方面的背景知识，文档开头对此也有简要介绍。

*2023*年*1*月*14*日，美国马萨诸塞州波士顿。

目录

1 器件可靠性基础 31.1 现实生活中的磁盘与相关研究.4

2 可靠性编码基础 5

3 一种具有并发维护的马尔可夫可靠性模型 7

3.1 持久性与平均数据丢失时间.12 3.2 案例研究：复制与纠删.12 3.3 平均数据丢失时间准确性的讨论.13 3.4 支持平均数据丢失时间和高级模型的论据.14 3.4.1 纳入硬错误（不可纠正错误）.14 3.4.2 针对非规则码或多个规则码的修改.15 3.4.3 使用最大距离可分码的二维存储阵列.19 3.4.4 基于异或的擦除码.22

4 广义马尔可夫模型的 MTTDL 23

4.1 MTTDL 的高效计算.23 4.2 案例研究：将广义模型应用于金字塔码.25

5 可用性 26

5.1 带超时的可用性.28 5.2 案例研究：谷歌不可用性数据.29

6 冷存储的多维马尔可夫模型 30

6.1 通用模型描述.31 6.2 特殊情况：三节点状态马尔可夫模型.34 6.3 冷存储中的故障类型和载体不可用性.36 6.4 转移率、概率矩阵和载体可用性.39 6.4.1 时变故障检测.40 6.4.2 时变载体修复.41 6.5 性能上下界.43 6.6 数据辅助建模框架.45 6.7 数值结果.48

7 高级建模 **52**

7.1 纳入扇区错误.53 7.2 模拟器.53

1 器件可靠性基础

全新产品投入使用后，在一段时间内能够令人满意地执行功能操作，这段时间称为有效使用期，之后最终发生故障，设备将无法再响应用户的请求。对于设备组件，观测到的失效时间（TTF）是一个连续随机变量，其概率密度函数为$f_{TTF}(t)$，表示产品直至首次发生永久失效的寿命。设备的失效概率可利用TTF的累积分布函数（CDF）计算如下：

$$F_{TTF}(t) \triangleq Pr\{TTF \le t\} = \int_0^t f_{TTF}(y)dy, \quad t > 0 \tag{1.1}$$

我们可以将$F_{TTF}(t)$视为时间 0 到x之间的不可靠度量。另一方面，可靠度函数$R(t)$定义如下：

$$R(t) \triangleq 1 - F_{TTF}(t) = \int_t^{\infty} f_{TTF}(y)dy \tag{1.2}$$

换言之，可靠性是时间 t 之前没有发生故障的概率，并通过式（1.2）与TFF的累积分布函数相关联。注意，式（1.2）表明$f_{TTF}(t) = -dR(t)/dx$。在现实中，可能无法直接根据可用的物理信息估计TFF的分布函数。为了阐明物理失效模式与TFF概率分布之间的关系，有一个有用的函数称为风险率 函数或故障速率 函数，通常记作$h_{TTF}(t)$。该函数定义如下：

$$h_{TTF}(t) \triangleq \frac{f_{TTF}(t)}{R(t)} = -\frac{dR(t)}{R(t)dt} \tag{1.3}$$

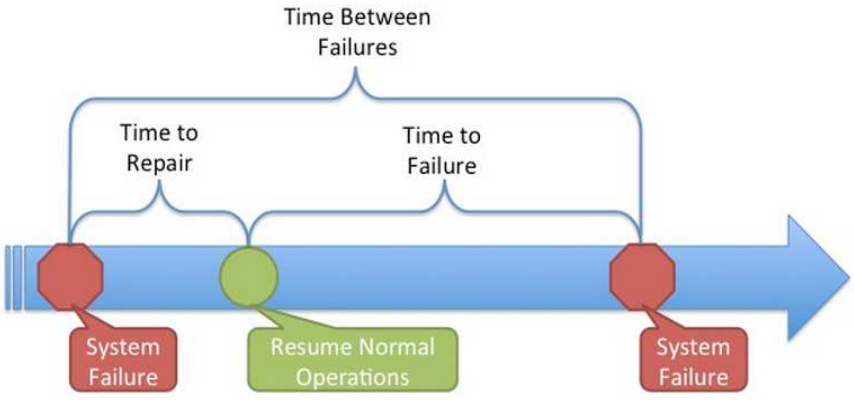


图 1.1：时域中平均失效与修复时间的示意图。

求解一阶常微分方程 (1.3)，可得如下关系

$$h_{TTF}(t) = -d(\ln(R(t)))/dt \tag{1.4}$$

初始条件为$R(0) = 1$。请注意，知道风险率就等同于知道分布。因此，谈到系统的可靠性时，我们会交替使用这些函数来量化可靠性。设备首次发生故障或数据完全丢失前的预期平均时间，是最常用的可靠性度量之一。平均故障时间（MTTF）定义为随机变量TTF的期望值，表示为

$$MTTF \quad = \quad \mathbb{E}[TTF] = \int_0^\infty R(t)dt \Leftrightarrow \lim_{t\to\infty} tR(t) = 0 \tag{1.5}$$

其中 $\mathbb{E}[.]$ 是期望算子。请注意，公式 (1.5) 对均值存在的分布均成立。为简化记号，在后续讨论中省略下标 TTF。

设备的年化故障率（AFR）常用于估算设备或组件在全年持续使用后发生故障的概率。传统方法假设从开始运行到发生故障的时间间隔相互独立，且服从速率为 λ 的指数分布。因此，AFR 表示为 $AFR = 1 - R(t) = 1 - e^{-\lambda t}$，其中 $\lambda = 1/MTTF$，且 x是以小时为单位的运行时间索引。平均故障时间以小时为单位报告，而一年有 8760 小时，因此 $AFR = 1 - e^{-8760/MT_{TF}}$。由于 $8760/MTTF \ll 1$，因此 $AFR \approx 8760/MTTF$。或者，若平均故障时间以年为单位表示，则 $AFR \approx 1/MTTF$。

1.1 现实中的磁盘及相关研究

在本文中，我们假设设备发生故障时，有外部人员可以对其进行修复。外部人员更换或修复故障设备所需的时间也是一个随机变量，其均值称为平均修复时间（MTTR）。为了避免等待平均修复时间，许多公司会购买备用产品／热插拔备件，以便快速安装替换设备。不过，客户通常会询问产品的维修周转时间，而这最终也可归入平均修复时间（MTTR）这一类别。最后，平均故障间隔时间（MTBF）是一项可靠性指标，用于表示同一设备或系统组件连续两次发生故障之间的时间 [1]。这是关于产品使用寿命最常见的咨询，也是最终用户决策过程中的重要依据。图 1.1 通过运行中的存储系统时间线概述了这些术语。

多篇研究论文表明，组件磁盘的平均更换率约为理论预测平均故障时间值的二十倍；也就是说，实际观察发现，预测的平均故障时间值偏低 [2]。研究还表明，磁盘故障率呈现如图 1.2 所示的“浴盆”曲线。此外，与传统的齐次随机模型不同，硬盘更换率不会进入稳态。使用几年后，观察发现驱动器（其中大多数是磁盘）会进入磨损期，在此期间，故障率随时间稳步上升。故障间隔时间

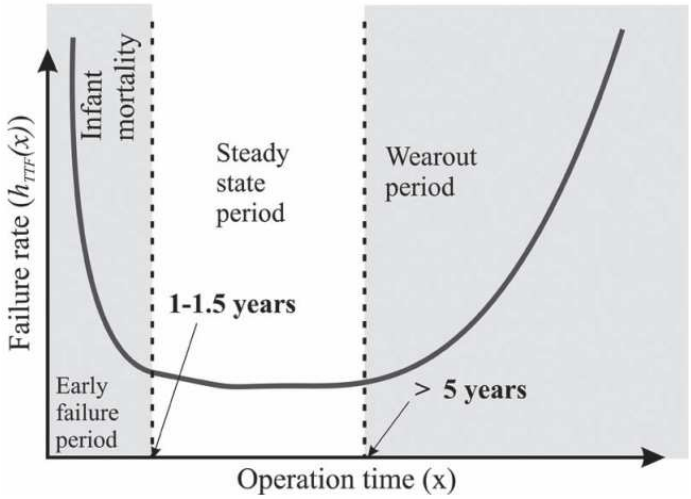


图 1.2：硬盘驱动器的风险率随运行时间变化的模式 [36] .

采用威布尔分布或伽马分布，比采用广泛使用的指数分布能更好地拟合这些数据 [2], [3]。大量证据表明，同一批次的磁盘故障之间存在显著相关性，而在许多应用中，这种相关性难以量化 [4]。

业界和学术界都投入了大量努力，试图通过平均故障寿命率准确预测大规模存储系统的可靠性。例如，文献 [5] 提出了一种准确但复杂的模型，将灾难性故障和使用相关的数据损坏纳入考量。作者特别指出，组件故障率与整个存储系统的故障率几乎没有关联，甚至可能毫无关联。据报告，尽管每个组件磁盘风险率都在上升，但连续两次系统故障之间的间隔时间仍比传统模型预测的要长得多 [6]。文献 [7] 引入并采用了磁盘擦洗，以应对通常与磁盘的容量、使用情况和运行状态无关的潜在缺陷。该研究还使用齐次泊松模型进行可靠性估算。文献 [8] 已充分论述并建模了磁盘故障寿命的重尾特性。一个最重要的趋势是利用收集到的数据（例如，Backblaze 提供了此类研究数据集 [9]），为预测性维护构建数据存储系统可靠性模型 [10]。在本文后文（第 6 节）中，我们将展示一个具体案例，说明如何利用数据对复杂系统建模，并借助传统数学工具估算存储系统的可靠性。

2 可靠性编码基础

在实际的数据存储系统中，会对数据节点或设备中包含的信息进行编码，以生成某种形式的冗余，从而增强用户数据抵御设备故障／缺陷或随机／突发错误的能力。每当存储节点／设备发生故障时，



图 2.1：一种简单的奇偶校验编码，在数据块末尾附加一个奇偶校验位，以确保 1 的总数为偶数。

系统控制器会识别这些设备故障，因此可以轻松确定故障位置。从编码的角度来看，这些故障被视为擦除。因此，生成这种冗余的过程称为擦除纠错编码，或简称为擦除码。擦除码利用系统化生成的冗余数据，提高整个存储系统的可靠性。错误／擦除纠错编码是一门相当古老且涵盖面广的学科，起源于近五十年前。在本文档的其余部分，我们将仅讨论针对数据存储系统设计的擦除码。

我们先简要介绍一下，以帮助理解后续内容。假设有n个设备，每个设备包含m个用户数据块。然后，我们使用擦除码生成冗余，将其划分为c个数据块，并将这些数据块存储在c个不同的设备中；也就是说，设备总数为$n = m + c$。一种空间最优（也称为最大距离可分（MDS））编码方案，能够根据任意m个n设备所包含的信息，重建m个设备的信息。因此可以看出，最大距离可分码支持一种规律且易于理解的恢复机制。最大距离可分码的一个简单实例是图 2.1 所示的基于 XOR 的奇偶校验编码 方案。采用“偶数”约定时，在数据块末尾再添加一个二进制位，使二进制 1 的总数保持为偶数。用线性代数的术语来说，对于由生成矩阵$\mathbf{G}$定义的一个（n，m）线性码，如果$\mathbf{G}$的所有$m \times m$子矩阵都可逆，那么该线性码就能容忍任意c个擦除。具有这一性质的码称为最大距离可分码。

我们可以将数据存储专用擦除码分为三类：扁平最大距离可分码、奇偶校验阵列码和扁平异或码。里德-所罗门码 [11] 是扁平最大距离可分码的一个例子，但在实际应用中，其实现的计算开销很大。大多数阵列码在设备数量方面是空间最优的，且计算开销低于里德-所罗门码，但通常只能容忍两到三台设备发生故障 [12]，[13]，[14]。由于本文档的主要目的是介绍这些码在可维护系统中的可靠性问题，因此不再介绍这些码的细节。关于存储应用的擦除码，一篇不错的入门教程见 [15]。

另一方面，大多数现代编码都是扁平的基于异或的码，用于保护存储应用中的数据。它们基于低复杂度、低局部性的[1]，不规则擦除纠错编码，即具有不规则故障容忍度的编码，具体包括低密度奇偶校验（LDPC）码 [16] 和喷泉（LT）码 [17]。这些编码的后续实现已考虑可修复性和更新约束，并提供了

[1]在存储擦除码的语境下，局部性指的是恢复特定存储单元或编码符号时需要访问的存储单元或编码符号的数量。局部性可以从纯编码的角度或编码+分配策略的角度来考量。所指的局部性语境会影响相关讨论，并可能得出不同结论。

有利于系统维护的结构 [18]。特别是，由于其不规则结构，这些编码已被证明能够在修复带宽与存储之间的权衡方面实现良好性能 [19], [20]。此类编码通常利用图论或多个代数构造的级联（乘积码） [21], 构建，通过将整体存储问题作为一个整体加以考虑来设计，并且需要对建模假设进行重大改变，从而引出了有关存储系统可靠性/持久性/可用性的有趣开放问题。此外，选择适当的编码来满足数据持久性约束，并不一定能解决整体可靠性问题，因为事实表明，在可用硬件和网络上进行资源分配 [22], [23] 以及编码数据的分配，对于整体可靠性至关重要且具有深远影响 [24]。这些以及类似的开放问题将在本文档后续章节中加以阐述和讨论。

3 并发维护下的马尔可夫可靠性模型

传统上，使用马尔可夫模型来评估纠删码存储系统的可靠性。由于我们关注系统故障或数据完全丢失的情形，因此所考虑的马尔可夫模型类别包含吸收态。吸收状态是指系统故障状态或数据完全丢失状态，一旦进入便无法离开。大多数模型都假设采用类似 RAID 的配置（即最大距离可分码——如前文所述，具备常规容错能力），故障之间相互独立，且故障时间和重建时间服从指数分布。对于这种情况，马尔可夫模型非常适合用于描述系统行为。事实上，尽管这些模型通常对实际系统的刻画极为简化，但它们仍为磁盘故障和修复对系统可靠性的影响提供了许多重要见解。

存储系统中使用的规范马尔可夫模型基于奇偶校验编码，它只能容忍n 个存储单元或设备中一个设备发生故障。模型共有三个状态：状态n 表示所有设备均处于活动且正常运行状态；状态$n-1$表示一个设备发生故障；状态F表示故障状态，因为至少有两个设备发生故障，且数据完整性已丢失。假设所有设备均以恒定速率 λ 发生故障或以恒定速率 μ得到修复。图 3.1 展示了该马尔可夫模型的状态图。我们先介绍以下引理，并利用其结果讨论停留时间，即在特定状态中停留所需的时间。

引理 1： 设 $X_1, \ldots, X_n$ 为相互独立的指数分布随机变量，其速率为 λ_i（$i = 1, \ldots, n$）。则 $\min\{X_1, \ldots, X_n\}$ 的分布函数为速率为 $\sum$ 的指数分布，且最小值为 X_s（$s \in \{1, \ldots, n\}$）的概率为 $\lambda_s / \sum$（i λ_i）。

证明：考虑以下概率：

$$Pr\{\min\{X_1, \ldots, X_n\} > t\} = \prod_i Pr\{X_i > t\} = e^{-(\sum_i \lambda_i)t} \tag{3.1}$$

这意味着，$\min\{X_1, \ldots, X_n\}$ 的累积分布函数是速率为 $\sum_i \lambda_i$ 的指数分布的累积分布函数。可以发现，$\max\{X_1, \ldots, X_n\}$ 并非如此。

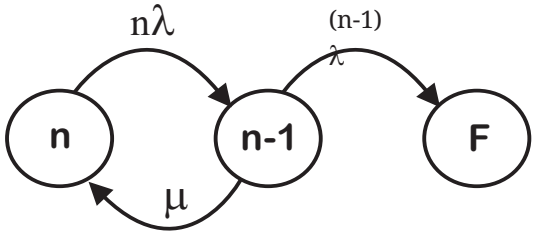


图 3.1：用于奇偶校验编码数据可靠性估算的典型连续时间马尔可夫模型。

此外，

$$Pr\{X_s\text{是最小值}\} = Pr\{X_s < Xj \text{ 对于 } j \neq s\} \tag{3.2}$$

$$= \int_0^\infty Pr\left\{X_s < X_j \text{ for } j \neq s | X_s = t\right\} \lambda_s e^{-\lambda_s t} dt \tag{3.3}$$

$$= \int_0^\infty \lambda_s e^{-\lambda_s t} \prod_{j \neq s} e^{-\lambda_j t} dt = \lambda_s \int_0^\infty e^{-(\sum_j \lambda_j) t} \tag{3.4}$$

$$= \lambda_s / (\lambda_1 + \cdots + \lambda_n) \tag{3.5}$$

如期望的那样。 □

由于在每个状态中，标签表示以相同速率λ发生故障的设备数量，因此从状态 n 转出的速率为 $n\lambda$。利用引理 1 的结果，我们可以得出，停留时间 Y_n 服从速率为 $n\lambda$的指数分布。类似地，在状态 $n-1$中的停留时间 Y_{n-1}服从速率为 $(n-1)\lambda+\mu$的指数分布。系统以概率$p_F = (n-1)\lambda/((n-1)\lambda+\mu)$从状态 $n-1$ 转移到状态 F（失效状态）的概率为.....

传统的存储系统可靠性分析使用一种称为平均数据丢失时间（MTTDL）的指标。MTTDL 是进入状态 F的预期（平均）时间。MTTDL 并非直接衡量可靠性的指标，而是基于可靠性得出的平均值，由 $\int_0^\infty R(t)dt$ 给出。对于描述现实生活中故障现象的大多数随机模型而言，MTTDL 可能很难计算。幸运的是，对于更简单的马尔可夫模型（例如图 3.1 所示的模型），有明确的数学方法可以计算 MTTDL。

假设在整个系统最终进入状态 $n-1$ 之前，我们反复访问状态 K 次。那么，数据丢失前的总时间为

$$Y_{DL} = \sum_{i=1}^{K} (Y_n^{(i)} + Y_{n-1}^{(i)}) \tag{3.6}$$

其中$Y(i)_n$是第i个停留时间。根据定义，MTTDL 为 $\mathbb{E}[Y_{DL}]$。以$K = k$为条件，并对所有可能情况求和（对K的分布进行无条件化——该分布

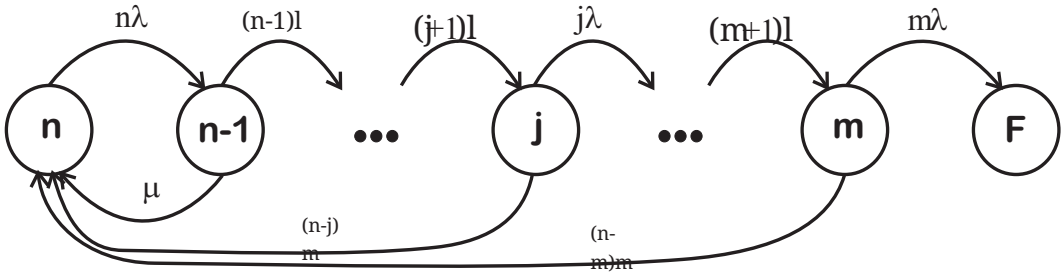


图 3.2：（n,m）编码数据可靠性估算的广义连续时间马尔可夫模型。

实际上是几何分布）即可得到

$$\begin{aligned}
\mathbb{E}[Y_{DL}] &= \sum_{k=1}^{\infty}\left(\sum_{i=1}^{k}(\mathbb{E}Y_n^{(i)}+\mathbb{E}Y_{n-1}^{(i)})\right)p_F(1-p_F)^{k-1} && (3.7)\\
&= \sum_{k=1}^{\infty}\left(\frac{k}{n\lambda}+\frac{k}{(n-1)\lambda+\mu}\right)p_F(1-p_F)^{k-1} && (3.8)\\
&= \frac{2n\lambda+\mu-\lambda}{n\lambda((n-1)\lambda+\mu)}\mathbb{E}[K]=\frac{2n\lambda+\mu-\lambda}{n\lambda((n-1)\lambda+\mu)}\frac{(n-1)\lambda+\mu}{(n-1)\lambda} && (3.9)\\
&= \frac{\mu+\lambda(2n-1)}{\lambda^2 n(n-1)} && (3.10)
\end{aligned}$$

请注意，Y_{DL} 的概率分布并非指数分布，但其均值很容易计算。事实上，Y_{DL} 的概率分布是指数分布之和，但难以用闭式计算。稍后我们将看到，对 Y_{DL} 的良好近似同样可以得到指数分布之和。我们将图 3.1 中的马尔可夫模型推广为：令系统能够纠正 c 个设备故障，并具有 m 个数据单元或设备，即 $n = m + c$。前述奇偶校验编码系统中 $c = 1$ 且 $m = n - 1$。此外，假设在每个状态下都进行并发修复，由此得到图 3.2 所示的马尔可夫模型。对于这种推广情形，可以采用基于拉普拉斯变换的技术计算平均数据丢失时间。

可靠度函数 $R(t)$ 表示在 t 时刻之前处于除 F 以外任意状态的概率。换句话说，

$$R(t) = P_n(t) + P_{n-1}(t) + \cdots + P_{m+1}(t) + P_m(t) \tag{3.11}$$

其中，$P_j(t)$ 表示在时间 t 处于 j 状态的概率。此外，对于 $t \geq 0$，可靠度函数 $R(t)$ 的拉普拉斯变换为

$$R^*(s) = \int_0^{\infty} R(t)e^{-st}dt \tag{3.12}$$

由方程（3.12）可知，MTTDL 是衡量可靠性的综合指标，

其值由变换后函数在 $s=0$ 处的取值给出，即:

$$MTTDL = R^*(0) = \int_0^\infty R(t)dt = \int_0^\infty P_n(t)dt + \cdots + \int_0^\infty P_m(t)dt \quad (3.13)$$

$$= P_n^*(0) + P_{n-1}^*(0) + \cdots + P_m^*(0) \quad (3.14)$$

利用一些概率论背景知识，我们将该模型在时域中的常规 Kolmogorov 方程写为如下形式 [25],

$$\begin{aligned} P_n'(t) + P_n(t).n\lambda &= P_{n-1}(t).\mu + P_{n-1}(t).2\mu + \cdots + P_{n-1}(t).(n-m)\mu \\ &\cdots \\ P_j'(t) + P_j(t).(j\lambda + (n-j)\mu) &= P_{j-1}(t).(j+1)\lambda \\ &\cdots \\ P_F'(t) &= P_m(t).m\lambda \end{aligned}$$

相应的变换域方程为:

$$\begin{aligned} sP_n^*(s) - P_n(0) + P_n^*(s).n\lambda &= P_{n-1}^*(s).\mu + P_{n-2}^*(s).2\mu + \cdots + P_m^*(s).(n-m)\mu \\ &\cdots \\ sP_j^*(s) - P_j(0) + P_j^*(s).(j\lambda + (n-j)\mu) &= P_{j-1}^*(s).(j+1)\lambda \\ &\cdots \\ sP_F^*(t) - P_F(0) &= P_m^*(s).m\lambda \end{aligned}$$

初始条件为 $P_n(0)=1$ 和 $P_j(0)=0$，其中 $j<n$。这些线性方程组很容易写成如下矩阵形式:

$$\begin{pmatrix} s+n\lambda & -\mu & -2\mu & \dots & -(n-m)\mu & 0 \\ -n\lambda & s+(n-1)\lambda+\mu & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & -m\lambda & s \end{pmatrix} \begin{pmatrix} P_n^*(s) \\ P_{n-1}^*(s) \\ \vdots \\ P_F^*(s) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

并可用矩阵记法表示

$$\mathbf{AP} = \mathbf{N}^{(0)}$$

由此，对矩阵 $(\mathbf{A}^{-1})$ 求逆并从右侧相乘，即可得到$\mathbf{P}$的解，也就是对于$j \in \{n, n-1, \dots, m, F\}$，有$P^*{}_j(s)$，其中$\mathbf{N}^{(l)} = [0\ 0 \dots 0\ 1\ 0 \dots\ 0\ 0]$，且第 $(l+1)$ 个向量分量为 1。当然，对于较大的n，可能需要采用高效的矩阵求逆方法。求得$P^*{}_j(s)$后，通过拉普拉斯逆变换可得到$P_j(t)$，最终得到$R(t)$。例如，在 [26]，中，针对$c=1$、$c=2$ 和$c=3$ 计算出的 MTTDL 如下

给定为

$$
\begin{aligned}
MTTDL_{c=1} &= \frac{\mu+\lambda(2m+1)}{\lambda^2 m(m+1)} \\
MTTDL_{c=2} &= \frac{2\mu^2+\mu\lambda(5m+6)+\lambda^2(3m^2+6m+2)}{\lambda^3 m(m+1)(m+2)} \\
MTTDL_{c=3} &= \frac{6\mu^3+\mu^2\lambda(17m+33)+\mu\lambda^2(14m^2+47m+33)+2\lambda^3(2m^3+9m^2+11m+3)}{\lambda^4 m(m+1)(m+2)(m+3)}
\end{aligned}
$$

由于我们关注的是矩阵逆的第一列，因此 MATLAB/Mathematica 的符号求逆计算功能可能非常有用，可用于计算合理选择c时的显式表达式。如$MTTDL_{c=1}$所示，这与我们先前推导的式 (3.10) 相同。我们注意到，当c较大时，MTTDL 表达式会变得相当复杂。然而，在传统存储系统中，$\omega=\mu/\lambda$ 比值非常大。因此，可以基于假设 $\omega=\mu/\lambda \gg 1$对 MTTDL 进行简化。在 [27], 中，基于 ω 比值非常大的假设，推导出了通用平均无数据丢失时间的近似表达式。该近似表达式相当简洁，如下所示

$$
MTTDL_c \approx \frac{\omega^c}{\lambda(n-c)\binom{n}{c}} \tag{3.15}
$$

遗憾的是，即使采用这一假设，当c取较大值时，$R(t)$ 的表达式仍会变得非常复杂。在 [26], 中，给出了适用于任意m和$c=0$、1、2 和 3 的$R(t)$良好近似，如下所示

$$
\begin{aligned}
R_{c=0}(t) &= e^{-m\lambda t} && (3.16)\\
R_{c=1}(t) &\approx C_{1,m,0}e^{-t/MTTDL_{c=1}} + C_{1,m,-\mu}e^{-t(\mu+\lambda(2m+1))} && (3.17)\\
R_{c=2}(t) &\approx C_{2,m,0}e^{-t/MTTDL_{c=2}} + C_{2,m,-\mu}e^{-t(\mu+\lambda(2m+3))} + C_{2,m,-2\mu}e^{-t(2\mu+\lambda m)} && (3.18)\\
R_{c=3}(t) &\approx C_{3,m,0}e^{-t/MTTDL_{c=3}} + C_{3,m,-\mu}e^{-t(\mu+\lambda(2m+5))} \\
&\quad + C_{3,m,-2\mu}e^{-t(2\mu+\lambda(m+1))} + C_{3,m,-3\mu}e^{-t(3\mu+\lambda m)} && (3.19)
\end{aligned}
$$

其中指数项的系数为

$$
C_{c,m,0} = 1 + \sum_{i=1}^{c} \frac{1}{i} m \binom{m+c}{c} \omega^{-c-1} \tag{3.20}
$$

and

$$
\begin{aligned}
&C_{1,m,-\mu} = -\frac{\lambda^2 m(m+1)}{\mu^2}, C_{2,m,-\mu} = -\frac{\lambda^3 m(m+1)(m+2)}{\mu^3}, C_{3,m,-\mu} = -\frac{\lambda^4 m(m+1)(m+2)(m+3)}{2\mu^4}, \\
&C_{2,m,-2\mu} = -\frac{\lambda^3 m(m+1)(m+2)}{4\mu^3}, C_{3,m,-2\mu} = -\frac{\lambda^4 m(m+1)(m+2)(m+3)}{4\mu^4}, \\
&C_{3,m,-3\mu} = -\frac{\lambda^4 m(m+1)(m+2)(m+3)}{18\mu^4}.
\end{aligned}
$$

经过繁琐的代数运算，这些表达式可以推广到任意m和c，尽管原始研究并未推导出这些表达式 [26]。一般情形下精确的闭式表达式仍是一个开放问题。

3.1 持久性与平均数据丢失时间

与可靠性理论相关的一项重要指标是持久性。该指标定义为系统能够为用户数据或实体提供访问的持续时间。如果数据受到冗余机制的保护，即复制或擦除纠错编码，那么持久性指的是编码或复制后的用户数据永久丢失，而不是奇偶校验块或副本块永久丢失。持久性的单位通常以天或年表示。不过，按照行业术语，持久性以几个9（9 的个数）表示。这指的是在系统运行的最初若干年内没有发生任何错误的概率。

如前所述，平均数据丢失时间这一综合性指标只能提供关于整个系统可靠性/持久性极少的信息。准确计算数据丢失时间分布能够提供系统可靠性的实际信息，但计算过程可能十分繁琐。我们已经看到，$R(t)$ 通常并不服从指数分布。不过，为简化起见，我们假设它服从速率为 $1/MTTDL$ 的指数分布。此时，9 的个数为 $\left\lfloor \log_{10}\left(\frac{1}{1-R(t)}\right)\right\rfloor$，其中 $R(t) = e^{-t/MTTDL}$ 是可靠性/持久性函数，而 $t \geq 0$ 是以平均数据丢失时间为单位表示的时间（通常以小时计）。例如，对于平均数据丢失时间为 2,500,000 小时、且关注运行时间为 1 年（8760 小时）的系统，其持久性数值计算如下：

$$R(8760) = e^{-8760/2500000} = 0.9965 \rightarrow \text{two 9's} \tag{3.21}$$

这意味着在使用的第一年、占空比为 100% 时，系统无故障运行的概率为 0.9965。接下来，我们将把这种简化的观点与上一节中针对复制冗余方案给出的近似进行比较。我们将在本文后面看到，指数近似对于持久性计算相当准确。

3.2 案例研究：复制与纠删

数据存储系统采用不同的冗余方案，以防止因多个设备同时发生故障而导致的数据丢失。复制是一种广泛使用的方案：每个数据块都会被复制，副本存储在不同的节点上，以提高多个存储节点发生故障时至少有一个副本幸存的可能性。在复制方案中，$m = 1$和$c = 1, 2, \ldots\ldots$ 表 3.1 表明，对于较大的 ω，针对$MTTDL$给出的近似，以及故障时间分布服从指数分布这一假设，都相当准确，尤其是在用几个9表示相关量时。

让我们将数据设备的数量从$m = 1$ 增加到$m = 100$，并重新计算$c = 1,2,3$时这些数值。结果如表3.2所示。可以看出，近似值

[2]这里所说的故障不一定要在同一时刻发生。我们将修复故障设备完成之前发生的额外故障称为并发故障。

故障/修复率		$c=1$		$c=2$		$c=3$	
λ	μ	$e^{-}t/MTTDL_1$	$Rc=1\ (t)$	$e^{-}t/MTTDL_2$	$Rc=2\ (t)$	$e^{-}t/MTTDL_3$	$Rc=3(t)$
1/200K	1/24	4	4	8	8	12	12
1/500千	1/24	5	5	9	9	14	14
1/1.2M	1/24	6	6	11	11	15	15
1/200K	1/240	3	3	6	6	9	9
1/500K	1/240	4	4	7	7	11	11
1/1.2M	1/240	5	5	9	9	12	12

表3.1：每个归档的3副本和4副本系统的九个九数量。

故障/修复率		$c=1$		$c=2$		$c=3$	
λ	μ	$e^{-}t/MTTDL_1$	$Rc=1(t)$	$e^{-}t/MTTDL_2$	$Rc=2\ (t)$	$e^{-}t/MTTDL_3$	$Rc=3(t)$
1/200K	1/24	1	1	3	3	5	5
1/500K	1/24	2	2	4	4	7	7
1/1.2M	1/24	2	2	5	5	8	8
1/200K	1/240	0	0	1	1	2	3
1/500K	1/240	1	1	2	2	4	4
1/1.2米	1/240	1	1	3	3	5	6

表3.2：每个归档的纠删码系统的九个九数量。

仍然大多准确；即使不准确，也属于低估值。

3.3 平均数据丢失时间准确性的讨论

尽管平均数据丢失时间是进行相对比较的有用工具，但对于绝对测量而言，它并无意义 [28]。此外，由于推导闭式表达式时采用了不切实际的假设，近来其有效性也受到质疑。系统设计人员可能更关注最初几年内的失效概率，而非平均故障时间。平均数据丢失时间的汇总性质通常只能向客户传达非常有限的信息。事实上，大多数客户关心的是$R(t)$，也称为持久性。

传统可靠性模型基于四个简化假设构建：唯一的故障类型是整设备故障；采用单磁盘容错码，例如奇偶校验码；设备以恒定速率发生故障；设备以恒定速率得到修复。传统马尔可夫模型虽然能够快速解析估计单磁盘容错系统的可靠性，但并不适用于多磁盘容错系统，也无法纳入扇区故障，且不能补偿故障和修复过程中的时间依赖性。

这组简单马尔可夫模型的假设存在若干问题。例如，许多模型没有考虑扇区故障，无法准确模拟设备重建过程，并且假设所有设备具有相同的故障/重建率。此外，要将其扩展为考虑m重纠删的一般模型并非易事。规范马尔可夫模型的到达间隔时间具有无记忆性，然而大多数情况下，人们假设如果另一个存储组件发生故障，已经处于重建过程中的存储组件会从头开始修复

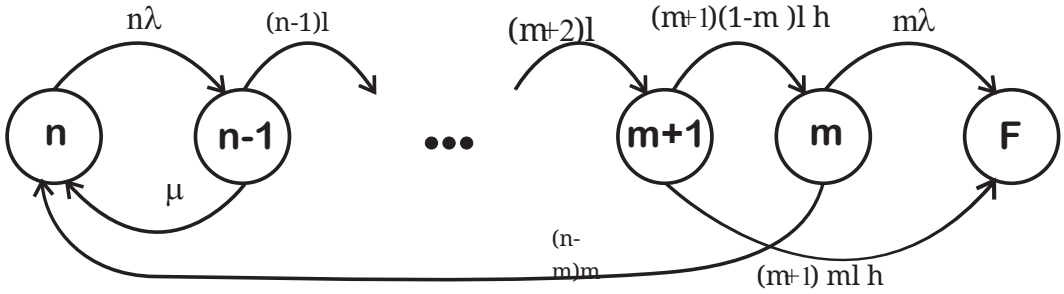


图3.3：用于估算（n,m）编码数据可靠性的连续时间马尔可夫模型，将硬错误纳入$MTTDL$估算。

。这些考量促使研究界探索更高级的建模技术，以更准确地估算可靠性。大多数研究强调推导解析结果的难度，并鼓励采用蒙特卡罗式仿真进行准确估算 [30]。

3.4 支持平均数据丢失时间及高级模型的论据

尽管先前对平均数据丢失时间的计算存在诸多不准确之处，但研究表明，这一广为人知的可靠性指标基于指数分布的故障和修复时间，只要组成存储系统的设备的平均故障时间远大于其平均修复时间 [29] ，且彼此独立运行，就不会受到故障/修复时间实际分布的影响。这一结果基本意味着，即使将设备故障分布从指数分布改为具有相同均值的其他分布，前几节中的系统平均数据丢失时间计算也不会受到影响。鉴于此，鲜有研究尝试让原始马尔可夫模型更贴近实际。本节将对此进行回顾，并提出用于可靠性估算的最一般形式的规范模型。

3.4.1 纳入硬错误（不可纠正错误）

当系统处于临界模式时，也就是再发生一次设备故障就会导致系统完全崩溃和/或数据丢失的状态时，可以发现这一要求是必要的。该要求使我们的马尔可夫模型略有变化，如图 3.3 所示。在此模型中， η 表示重建过程中每次读取设备时出现不可纠正错误的概率。令 UCER 表示设备的不可纠正错误率（例如 10^{-15}，以每读取一定数量的字节或比特所发生的错误数表示），则 η 为设备容量 × UCER。尽管这是已发表的表达式 [31],，但其中存在一个不准确之处：设备容量 × UCER 可能大于 1。然而， η 似乎表示一个概率项，因此必须满足 $0 \le \eta \le 1$。因此，更准确的表达式可写为$\eta = 1 - (1 - \text{UCER})$设备容量。

引入从状态 $m+1$ 到状态 F的转移，用于对系统在读取和重建t个设备故障时遭遇不可纠正错误的速率进行建模。根据 [31], 中的分析，该速率计算为磁盘发生故障的速率

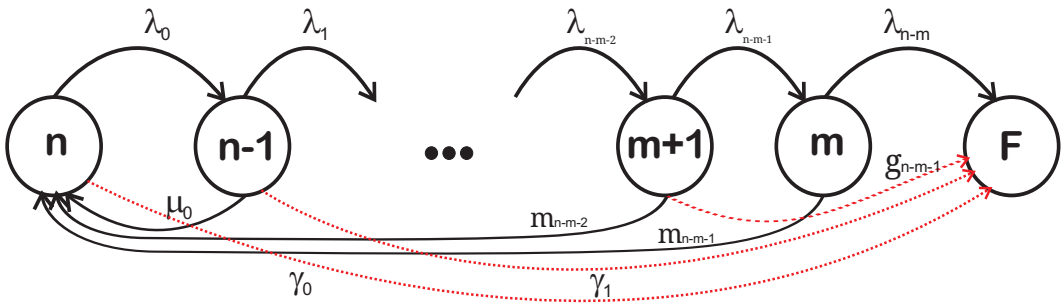


图3.4：一种广义连续时间马尔可夫模型，可用于估算包含不规则容错能力和硬错误的n、m）编码数据的$MTTDL$可靠性。

当有（m+1）个设备可用时，即（m+1）λ，以及读取m 个设备以进行重建时遭遇不可纠正错误的概率（注意，这里假设使用传统的最大距离可分码，它可能需要读取许多设备），即$P_{UCER}=m\eta$。与前面的论证类似，该乘积可能大于 1。更准确的表达式如下：

$$P_{UCER} = 1-(1-\eta)^m \tag{3.22}$$

逐设备修复策略下 MTTDL 的近似解（未考虑针对 η 和 P_{UCER}所作的修正，即 η = 设备容量 × 不可纠正错误率 和 $P_{UCER} = m\eta$）见 [31]。运用类似的论证，我们可以轻松地将这些结果推广到并发修复，并得到以下关于 $\eta \ll n$和 $\omega = \mu/\lambda \gg 1$的表达式，

$$MTTDL_c \approx \frac{c!w^c}{n(n-1)\ldots(n-c)(\lambda+\eta\mu)} \tag{3.23}$$

3.4.2 非规则码或多个规则码的修改

除了前述改进之外，我们将模型推广到更常见的非规则纠删码类别。“非规则”指的是基于二分图构造、具有非规则容错能力的现代非最大距离可分纠删码。此类中最具代表性的是基于 XOR 的 LDPC 码或喷泉码，它们具有良好的可修复性，且编码和解码操作简单。

下面介绍一种最一般的马尔可夫模型，以及其在采用非规则码保护的系统可靠性分析中的应用。请注意， [31], 中进行了少量推广，但并未达到图 3.4 所示的程度。此外，还考虑并分析了该模型的一些特殊情况，例如 [32]。在此模型中，设备故障以速率 λ_i一次发生一个；由于纠删码具有不规则容错能力，因此除状态 m之外，还可能以某个速率从任意状态转移到失效状态。此外，正如以往研究中所述，马尔可夫模型假定忽略重建时钟。换言之，每当另一块磁盘在重建期间发生故障时，每个马尔可夫模型都会为所有正在重建的磁盘重新计时重建时间。

发生故障。本文考虑的并发重建模型采用一种重建策略：每当一块磁盘发生故障时，就重新启动所有故障磁盘的重建。或者，也可以在每个状态下估计剩余修复时间，即随着故障数量增加，持续提高正在进行的重建操作的速率。因此，在一般模型中，我们应允许使用不同的重建速率 μ_i，以更好地模拟现实世界。

关于这个一般模型，有三个问题需要探讨：

- 鉴于非规则码的特性及其容错能力，如何高效计算速率 $\{\lambda_i, \gamma_i\}$，以便将其用于该模型？
- 鉴于重建的特性以及相关的系统格式／结构，如何估算速率 $\{\mu_i\}$，以便将其用于该模型？
- MTTDL 是否存在闭式表达式？$R(t)$ 呢？对于任意 c 和 m，情况如何？

我们先从第一个问题开始。这个问题实际上已在 [31] 中得到解答。给定非规则码和符号分配策略（例如 CRUSH [33]），我们将概率 p_k 记为：在系统已经容忍 k 个设备故障的前提下，整个系统还能再容忍一次设备故障的概率。从运行（存活）状态转移到故障（吸收）状态有两种情形。首先，在第 j 个状态（$n \geq j \geq m$）下，又有一个设备发生故障，而纠错码无法容忍这次损失，状态变化最终导致系统进入故障状态。这种情况发生的速率为 $j\ \lambda(1-p^{n-j})$，这可直接根据概率论中的基本马尔可夫模型相乘得到。另一种导致数据丢失的情形是：又有一个设备发生故障，擦除码能够容忍这一故障，但在重建过程中遇到硬错误（不可纠正错误），而擦除码无法容忍这次额外错误。这种情况发生的速率为 $j\ \lambda p^{n-j}(1-p^{n-j+1})(j-1)\eta$，可通过条件概率推导得到。当然，这里假设所有运行中的 $j-1$ 个设备在重建过程中都会被使用／访问；对于再生擦除码 [?]，情况未必如此。因此，可以根据所用擦除码的特性修改该表达式。最后，由于 $\lambda_{n-j}+\gamma_{n-j}=j\lambda$，当 $n \geq j \geq m+1$ 时，我们有如下结果

$$\gamma_{n-j} = j\lambda\left((1-p_{n-j})+p_{n-j}(1-p_{n-j+1})(j-1)\eta\right) \tag{3.24}$$

$$\lambda_{n-j} = j\lambda-\gamma_{n-j} \tag{3.25}$$

其中 $\lambda_{n-m}=(n-m)\lambda$ 和 $\gamma_{n-m}=0$。

这里，我们仍需说明如何获得条件概率p_ks。我们将系统整体能够容忍k次设备故障的无条件概率记为q_k，其中总共有$\binom{n}{k}$种可能情形。假设其中s_k种情形可被系统容忍，则有$q_k = s_k \Big/ \binom{n}{k}$。假设系统能够容忍一种$k+1$故障模式，那么其中任意大小为$k$的故障子集都可以被容忍。这意味着$p_k = q_{k+1}/q_k$，更明确地说，

$$p_k = \frac{s_{k+1}}{\binom{n}{k+1}} \Big/ \frac{s_k}{\binom{n}{k}} = \frac{s_{k+1}(k+1)}{s_k(n-k)} \tag{3.26}$$

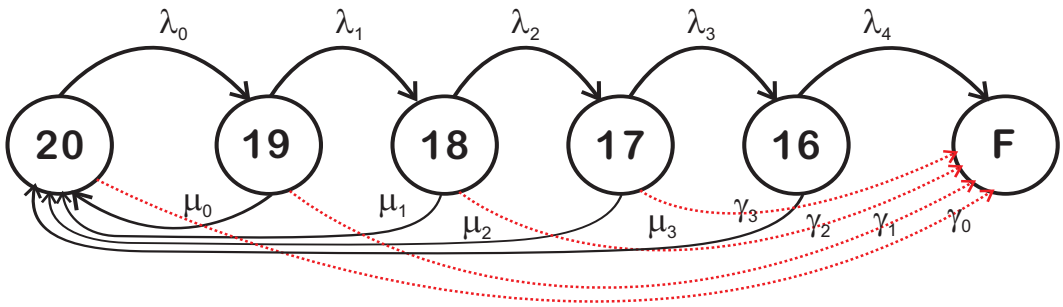


图 3.5：用于估算两个 (10, 8) RAID 6 阵列可靠性的广义马尔可夫模型。

其中，s_ks 通常通过对受不规则纠删码保护的整个系统进行复杂测试（擦除码性能/属性以及符号分配策略）来确定。无论分配策略如何，[34] 等研究都会考察线性分组码的最小擦除模式等容错指标。正如我们即将看到的，这类研究的结果有助于计算纠删码存储系统的持久性数值。

对于第二个问题，我们可以简单地假设 $\mu_i = \mu$（即采用同质修复过程），其中，$n - i$ 台设备的重建时间与单台设备的重建时间相同。当然，更复杂的方法会提高最终持久性结果的准确性。在另一种系统中，并发故障修复可能意味着 $\mu_i = (i+1)\mu$（即渐进式修复），因为每台设备都以相同的速率 μ进行修复。因此，修复率的选择很大程度上取决于整个系统的实现方式。

对于第三个问题，我们可以使用 Mathematica 或 Matlab 符号工具箱，计算较小m 和c值对应的闭式表达式。推导适用于任意m 和c 的闭式表达式，主要有两个重要原因。首先，我们可以对长分组长度非规则码的持久性进行建模（这类码的分组长度在实际设计中通常远大于传统代数码）。其次，我们可以对多种（实际上是大量）短分组长度的最大距离可分保护的 RAID 类型系统进行建模。

举例说明后一种情况：考虑两个 2-out-of-10（$m = 8$，$c = 2$）RAID6 系统，它们使用 $n = 20$ 来存储用户数据。我们注意到

- 所有零设备故障都可以容忍（$q_0 = 1$）。
- 所有单设备故障（共 $\binom{20}{1} = 20$ 种可能）都可以容忍（$q_1 = 1$）。
- 所有双设备故障（共 $\binom{20}{2} = 190$ 种可能）都可以容忍（$q_2 = 1$）。
- 在 $1140 = \binom{20}{3}$ 种三设备故障中，只有 900 种可以容忍（$q_3 = 0.7895$）。
- 在 $4845 = \binom{20}{4}$ 种三设备故障中，只有 2025 种可以容忍（$q_4 = 0.4180$）。
- 无法容忍超过四个设备同时发生故障（$q_k = 0$ 对于 $k > 4$）。

故障/修复率		通用马尔可夫（HR）		通用马尔可夫（PR）		传统方法（PR）	
λ	μ	平均数据丢失时间	耐久性九个9	平均数据丢失时间	耐久性九个9	平均数据丢失时间	耐久性九个9
$1/200K$	$1/24$	1.035×10^{9}	5	1.1×10^{9}	5	1.93×10^{10}	6
$1/500K$	$1/24$	6.9×10^{9}	5	7.1×10^{9}	5	3.01×10^{11}	7
$1/1.2M$	$1/24$	4.1×10^{10}	6	4.13×10^{10}	6	4.17×10^{12}	8

表 3.3：具有 $m = 8$ 和 $c = 2$的两个 RAID6 系统的九个九数量。HR：同质修复，即 $\mu_i = \mu$；PR：渐进式修复，即 $\mu_i =$ （$i + 1$）μ。

由此我们推得 $\{p_0, p_1, p_2, p_3, p_4\} = \{1, 1, 0.7895, 0.5294, 0\}$。图 3.6 所示的马尔可夫模型用于根据以下参数集估算可靠性，

- $\gamma_0 = n\lambda\left((1-p0) + p0(1-p1)(n-1)\eta\right) = 0,$
  $\lambda_0 = n\lambda - \gamma_0 = n\lambda = 20\lambda.$
- $\gamma_1 = (n-1)\lambda\left((1-p1) + p1(1-p2)(n-2)\eta\right) = 0.2105\times(n-1)(n-2)$
  $\lambda\eta = 72\lambda\eta, \lambda_1 = (n-1)\lambda - \gamma_1 = 19\lambda - 72\lambda\eta.$
- $\gamma_2 = (n-2)\lambda\left((1-p2) + p2(1-p3)(n-3)\eta\right) = 3.79\lambda - 33.4\lambda\eta,$
  $\lambda_2 = (n-2)\lambda - \gamma_2 = 18\lambda - 3.79\lambda + 33.4\lambda\eta = 14.21\lambda + 33.4\lambda\eta.$
- $\gamma_3 = (n-3)\lambda\left((1-p3) + p3(1-p4)(n-4)\eta\right) = 8\lambda - 144\lambda\eta$
  $\lambda_3 = (n-3)\lambda - \gamma_3 = 17\lambda - 8\lambda + 144\lambda\eta = 9\lambda + 144\lambda\eta.$
- $\gamma_4 = 0\lambda_4 = (n-4)$
  $\lambda = 16\lambda.$

进一步假设我们使用容量为 $1TB$ 的磁盘作为存储设备，其 10^{-15} 不可修复错误速率为 $\eta = 10^{-3}$。通过 MATLAB 进行符号计算，得到以 $9's$ 为单位的平均数据丢失时间以及持久性结果，并在表 3.3 中进行了评估和展示。下一节将给出这一一般情形下的闭式表达式，其中m和c可用于实现高效计算。

可以看出，改变修复策略对平均数据丢失时间结果的影响很小。此外，还列出了使用每个 RAID6 阵列的基本模型所得到的结果，并利用公式（3.15）计算平均数据丢失时间。该模型假设数据丢失概率分布为指数分布，因此基于数据丢失率的简单代数相加，计算两个 RAID6 阵列的平均数据丢失时间。这种传统方案同样没有考虑不可纠正错误。正如预期，采用传统方案得到的平均数据丢失时间和持久性数值更加乐观。

对于任意数量的由设备组成的 MDS 保护阵列，结果表明，s_k存在一个闭式表达式。换言之，假设我们有 π 个阵列，每个阵列包含 n 个设备，其中 c 个设备用于故障恢复。从 πn 个设备中选取 k 个故障设备的总选法数为$\binom{\pi n}{k}$。进一步假设，有 rj 个阵列

其中有 j 个设备发生故障，且 $0 \le j \le c$。s_k 的一般表达式可写为 [31]

$$s_k = \sum_{\substack{r_0+r_1+\cdots+r_c=\pi \\ \sum_{i=0}^{c} i r_i = k}} \binom{\pi}{r_0, r_1, \ldots, r_c} \prod_{i=0}^{c} \binom{n}{i}^{r_i} \tag{3.27}$$

这个表达式成立，是因为

- 多项式系数表示将i个故障分配到r_i 个阵列等情况的组合数。
- 已知有r_i 个阵列发生了i 个故障，分配这些故障的方法有$\binom{n}{i}_{r_i}$种，

约束条件是阵列总数必须为 π ，故障数必须为k。多项式定理是二项式定理的推广。这个推广版本意味着我们有

$$(x_0 + x_1 + \cdots + x_c)^{\pi} = \sum_{r_0+r_1+\cdots+r_c=\pi} \binom{\pi}{r_0, r_1, \ldots, r_c} \prod_{i=0}^{c} x_i^{r_i} \tag{3.28}$$

若将x_i替换为$\binom{n}{i} x^i$，则有

$$\left(1 + \binom{n}{1} x + \binom{n}{2} x^2 + \cdots + \binom{n}{c} x^c\right)^{\pi} = \sum_{r_0+r_1+\cdots+r_c=\pi} \binom{\pi}{r_0, r_1, \ldots, r_c} \prod_{i=0}^{c} \binom{n}{i}^{r_i} x^{\sum_{i=0}^{c} i r_i} \tag{3.29}$$

由此可知，若令 $\sum^{c}{}_{i=0}\, i r_i = k$，我们便可得出$s_k$是该展开式中$x^k$的系数，即，

$$s_k = coef\left(\left(\sum_{i=0}^{c} \binom{n}{i} x^i\right)^{\pi}, x^k\right) \tag{3.30}$$

因此，我们有

$$p_k = \frac{coef\left(\left(\sum_{i=0}^{c} \binom{n}{i} x^i\right)^{\pi}, x^{k+1}\right)(k+1)}{coef\left(\left(\sum_{i=0}^{c} \binom{n}{i} x^i\right)^{\pi}, x^k\right)(\pi n - k)} \tag{3.31}$$

3.4.3 使用最大距离可分码的二维存储阵列

另一个值得关注的情形是在二维存储阵列中使用最大距离可分码。过去的研究表明，与单维方案相比，某一类二维 RAID 方案具有更高的可靠性 [35]。然而，尚未开展一般性研究。近期研究如 [36], 仅在独立性等特定假设下，考察多维 RAID 型存储系统的平均首次故障时间。本文则对二维阵列进行更一般的研究，并旨在利用一般马尔可夫模型估算 MTTDL 的可靠性。为此，我们需要确定二维存储阵列的s_ks 及错误容忍能力。我们假设

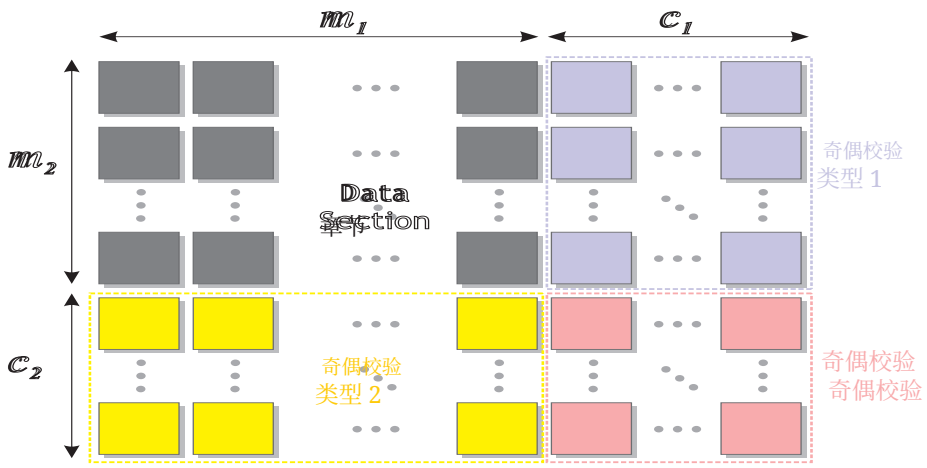


图 3.6：一种二维 RAID 方案，其中m_1m_2个数据块由总计$c_1k_2+c_2k_1+c_1c_2$个校验块保护。

每个水平和垂直阵列组件分别由具有c_1个和c_2个校验块的最大距离可分码保护。假设使用简单的有界距离译码器。适当时，可以沿水平方向或垂直方向开始解码。为简化符号，令$n_1=m_1+c_1$，且$n_2=m_2+c_2$。对于s_k，我们有以下猜想：若$k\le(c_1+1)(c_2+1)+\min\{c_1,c_2\}$

$$s_k=\begin{cases}\binom{n_1n_2}{k} & \text{if } k\le c_1c_2+c_1+c_2\\ \binom{n_1n_2}{k}-\binom{n_1}{c_1+1}\binom{n_2}{c_2+1}\binom{n_1n_2-(c_1+1)(c_2+1)}{k-(c_1+1)(c_2+1)} & \text{if } (c_1+1)(c_2+1)\le k\le(c_1+1)(c_2+1)+\min\{c_1,c_2\}\end{cases}$$

当s_k适用于$k>(c_1+1)(c_2+1)+\min\{c_1,c_2\}$时，其精确表达式会变得更加复杂，而任意$k$对应的闭式表达式仍是一个开放问题。我们来看一个使用最大距离可分码的二维存储阵列的特例。

示例 1：镜像（n_1,m_1）编码阵列（$c_1=n_1-m_1$个奇偶校验），是二维存储阵列的一种特殊情况，其垂直编码仅通过信息复制进行单次校验生成。例如，RAID 51 和 RAID 61 系统就是此示例的特殊情况。要使一列发生故障，两个副本都必须不可用/故障。对于二维阵列中给定的 k 个设备故障，列故障最多为 $\lfloor k/2\rfloor$。由于水平阵列最多可容忍 c_1 个设备故障，为了使整个系统能够恢复数据，列故障的最大数量不能超过 $\min\{\lfloor k/2\rfloor, c_1\}$。对于给定的 $0\le j\le\min\{\lfloor k/2\rfloor, c_1\}$ 个列故障，我们可以看到，利用 $k=k-2j$、$n=2$ 以及 $r=n_1$，即可使用公式（3.30）计算每个 j 对应的可容忍情形数量。还需注意，每种情形构成一个互不相交的集合。由于这些列有 $\binom{n_1}{j}$ 种可能的选取方式，因此我们有

$$s_k=\sum_{j=0}^{\min\{\lfloor k/2\rfloor,c_1\}}\binom{n_1}{j}coef\left((1+2x)^{n_1},x^{k-2j}\right)\mathbf{1}_{y\le n_1+j}(k) \tag{3.32}$$

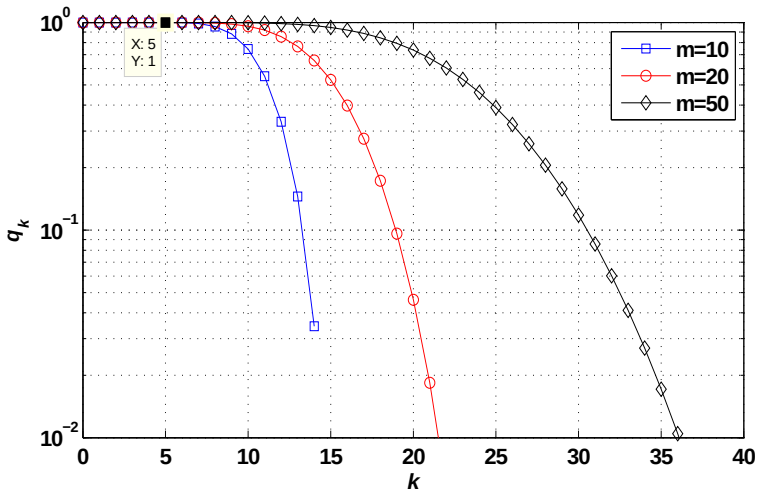


图 3.7：q_k 值随 k 和 m 变化的情况，对应 RAID61 配置。

其中，$\mathbf{1}_{y\le A}(y)$ 是示性函数；当 $y \le A$ 时，其值为 1。我们的表达式中需要该函数，因为对于求和中的每一项，我们都假设有 j 个列故障；如果 $k-2j > n_1 - j$ 或 $k > n_1 + j$，这意味着至少有 $j+1$ 个列故障，这与我们假设有 j 个列故障的条件相矛盾。

图 3.7 展示了 RAID61（$c_1 = 2$）配置下，q_k 随 k 变化的计算结果，其中 m_1 =分别取 10、20、50，且 $n_2 = 2$。请注意，对于每种情况，如果 $k \le c_1 c_2 + c_1 + c_2 = 5$，那么所有由 k 个故障设备构成的组合都可以容忍。例如，在保持 $c_2 = 1$ 不变的情况下，使用奇偶校验编码推导 $n_2 = 3$ 的类似表达式，或许会很有意思。即使是这个简单的扩展，表达式也可能变得相当复杂。我们还注意到，对于该特定配置，要计算 s_k，所有设备上的解码算法的精确定义也起着关键作用。例如，可以允许垂直和水平解码迭代继续进行，直到解码无法再改善故障统计为止。

练习1： 利用上述示例的结果，计算 RAID51、RAID6 和 RAID61 的平均数据丢失时间估计值，以便进行比较。计算时使用 $\lambda = 1/200K$ 和 $\mu = 1/24$。

利用这一结果，可以轻松将其推广到任意 n_2、具有 $m_2 = 1$ 和 $c_2 = n_2 - 1$ 的情形。该推广的结果给出了一个闭式表达式，如下列引理所示。

引理2： 对于仅有一个约束 $m_2 = 1$ 的存储设备二维阵列，s_k 的闭式表达式为

$$s_k = \sum_{j=0}^{\min\{\lfloor k/n_2 \rfloor, c_1\}} \binom{n_1}{j} coef\left(\left((1+x)^{n_2} - x^{n_2}\right)^{n_1}, x^{k-n_2 j} \right) \mathbf{1}_{y \le n_1 + c_2 j}(k) \tag{3.33}$$

证明：利用前面关于 $c_2 = 1$ 和 $n_2 = 2$ 的示例中的论证，可以很直接地证明这个引理。□

不受引理2约束时，s_k的闭式表达式会变得更加复杂。事实上，这正是前面提到的那个有趣的组合问题。尽管过去已有许多研究利用球与箱子论证来计算这类量，但对于本框架所假设的特定解码原理（在水平和垂直两个方向上迭代），目前似乎尚未发表适用于一般情况的解法。

3.4.4 基于异或的擦除码

对于任意擦除码——可能是由非最大距离可分码保护的设备阵列，计算s_k颇具挑战性，且在很大程度上取决于该码的图结构。即使不考虑编码符号分配方法，确定码的容错性也可能相当复杂。例如，文献 [37] 提出了一种用于纠删的线性码（基于异或的码）构造方法，该码采用系统二进制生成矩阵。事实证明，同一算法也适用于非系统码。该论文的论述基于码的可解码性，即研究假定使用最优解码器。实际上，这种容错性概况取决于该码所采用的编码和解码算法。

要完整刻画基于异或的擦除码的容错性，必须枚举每一种擦除模式集合。遗憾的是，需要枚举的擦除模式数量呈指数级增长。因此，不去找出所有会导致解码器故障的擦除模式，而是通过最小擦除集来刻画擦除码的容错性。最小擦除集是指一组会导致数据无法恢复而丢失的擦除，其中每个擦除对于造成这种结果都是必要且充分的。此外，所有最小擦除集的枚举称为最小擦除集列表（MEL）。与所有会导致数据丢失的擦除模式相比，MEL 的规模相对较小，却能完整刻画擦除码的容错性。文献 [37] 提出了一种算法[3]，用于高效确定由二进制生成矩阵定义的线性基于异或的擦除码的 MEL。给定一个（n，k）线性基于异或的码的 MEL，最小擦除向量（MEV）的长度为$n-k$，其中第i项统计权重为i的最小擦除模式数量，适用于$1 \le i \le n-k$。不难看出，MEV 有助于计算s_k。

示例 2：考虑以下用于 (8,4) 系统化 XOR 线性码的二进制生成矩阵，

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

使用 ME 算法，我们可以计算 MEV $=\{0, 0, 4, 5\}$。由此可见，$s_3 = \binom{8}{3} - 4 = 52$，并且通过简单枚举，$s_4 = \binom{8}{4} - 25 = 45$和$s_k = 0$适用于所有$k \ge 5$。

[3]该算法的 MATLAB 实现可在 http://www.suaybarslan.com/mea.txt 找到

4 广义马尔可夫模型的平均数据丢失时间

本节将给出上一节介绍的最一般马尔可夫模型的 MTTDL 精确表达式和近似表达式。本节介绍的内容部分发表于 [38]。如前所述，当c、m的取值较大，且存储阵列规模较大时，传统计算方法便不再适用。闭式表达式在这方面至关重要，能够用于预测大型存储阵列以及采用不规则现代编码技术编码的海量数据阵列的未来可靠性估计。

4.1 平均数据丢失时间的高效计算

按照我们之前的构造，变换域中的 Kolomogorov 方程被写入如下所示的系数矩阵 $\mathbf{A}$ 中。

$$\mathbf{A}=\begin{bmatrix} s+\lambda_0+\gamma_0 & -\mu_0 & -\mu_1 & \cdots & -\mu_{c-1} & 0 \\ -\lambda_0 & s+\lambda_1+\mu_0+\gamma_1 & 0 & \cdots & 0 & 0 \\ 0 & -\lambda_1 & s+\lambda_2+\mu_1+\gamma_2 & \cdots & 0 & 0 \\ 0 & 0 & -\lambda_2 & \cdots & 0 & 0 \\ \vdots & \vdots & & \ddots & 0 & 0 \\ 0 & 0 & \cdots & s+\lambda_{c-1}+\mu_{c-2}+\gamma_{c-1} & 0 & 0 \\ 0 & 0 & \cdots & -\lambda_{c-1} & s+\lambda_c+\mu_{c-1} & 0 \\ -\gamma_0 & -\gamma_1 & \cdots & -\gamma_{c-1} & -\lambda_c & s \end{bmatrix}$$

可以看出，当矩阵规模较大时，使用符号软件工具箱求 $\mathbf{A}$ 的逆可能颇具挑战。相反，利用系数矩阵的特殊结构，可以得到简洁的闭式表达式。我们先不加证明地陈述本节的关键结果。对于给定数量的冗余 c 个单元/设备以及 $x=0,\ 1,\ldots,c$，定义如下关键速率向量及其对应的各项，

$$\mathbf{\Lambda}_x^{(c)} \triangleq \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{c-1} \\ \lambda_c \end{bmatrix} + \mathbf{V}_x^{(c)} \begin{bmatrix} \gamma_0 \\ \mu_0+\gamma_1 \\ \mu_1+\gamma_2 \\ \vdots \\ \mu_{c-2}+\gamma_{c-1} \\ \mu_{c-1} \end{bmatrix}$$

其中

$$\mathbf{V}_x^{(c)} = \begin{bmatrix} \mathbf{0}_x & \mathbf{0}_{x\times c+1-x} \\ \mathbf{0}_{c+1-x\times x} & \mathbf{I}_{c+1-x} \end{bmatrix}$$

以及$\mathbf{I}_{c+1-x}$和$\mathbf{0}_x$分别为单位矩阵和全零矩阵。设$\Lambda_x^c(j)$表示向量$\mathbf{\Lambda}_x^{(c)}$的第$(j+1)$个元素，其中$0\le j\le c$。那么，我们得到以下变换

域表达式，其取值点为$s = 0$,

$$P^*_{m+c-x}(0) = \frac{1}{\phi_c(0)} \prod_{\substack{j=0 \\ j\neq x}}^{c} \Lambda^c_x(j) \tag{4.1}$$

其中，分母可以通过以下递推方式计算

$$\phi_t(0) = \prod_{i=0}^{t} \lambda_i + (\mu_{t-1} + \lambda_t) \left[\phi_{t-1}(0) - \prod_{i=0}^{t-1} \lambda_i + \gamma_{t-1} \left(\prod_{i=0}^{t-2} (\gamma_i + \lambda_i) + \xi_t \right) \right] \tag{4.2}$$

对于 $1 \leq t \leq c$，初始条件为 $\phi_0(0) = \lambda_0$。这里，我们可以证明 $\xi_1 = \xi_2 = 0$ 和$\xi_3 = \gamma_0\mu_0$。当t较大时， $\xi_{\{t>3\}}$ 的精确表达式会变得更加复杂。不过，我们注意到，与给出的 $\phi_t(0)$表达式的其余部分相比， ξ_s相对较小。因此，一个较好的近似为

$$\phi_t(0) \approx \prod_{i=0}^{t} \lambda_i + (\mu_{t-1} + \lambda_t) \left[\phi_{t-1}(0) - \prod_{i=0}^{t-1} \lambda_i + \gamma_{t-1} \left(\prod_{i=0}^{t-2} (\gamma_i + \lambda_i) \right) \right] \tag{4.3}$$

利用上述论证，我们可以求得$MTTDL_c$的闭式表达式，如下所示，

$$MTTDL_c = \sum_{x=0}^{c} P^*_{m+c-x}(0) = \frac{1}{\phi_c(0)} \sum_{x=0}^{c} \prod_{\substack{j=0 \\ j\neq x}}^{c} \Lambda^c_x(j) \tag{4.4}$$

如果使用方程 (4.3) 近似 $\phi_c(0)$，方程 (4.4) 将变为近似式（实际上是一个高估值）。例如，我们对 $\lambda = 1/200K$、 $\mu = 1/24$、$m = 8$、 $\eta = 10^{-3}$、$c = \{1, 2,3,4,5\}$ 和 $r = \{1, 2, 3\}$ 执行了实际计算和近似计算，观察到近似造成的误差小于 10^{-6}。我们希望为 ξ_t 的演变求得紧确界，并预测可能引入计算中的最大误差。对于一种特殊情况， ξ_t 可以表示为闭式，如以下引理所述。

引理 3： 对于 $t \geq 4$ 和 $x \leq c - 2$，如果多个最大距离可分保护阵列或一个基于异或的擦除码存储阵列能够容忍 x 个或更少的设备故障，则有 $\gamma_0 = 0$，$\ldots$， $\gamma_{x-2} = 0$因此

$$\xi_t \begin{cases} = \gamma_{t-3}\mu_{t-3} \prod_{i=0}^{t-4} \lambda_i & if\ t \leq x+2 \\ > \gamma_{t-3}\mu_{t-3} \prod_{i=0}^{t-4} \lambda_i & Otherwise. \end{cases} \tag{4.5}$$

证明：在建立ξ_t作为底层马尔可夫过程中的故障率和修复率的函数的演化过程时，我们将给出该引理的证明。□

请注意，这一结果意味着，如果根据编码算法和分配策略，任意$c - 2$ 个或更多设备故障的组合都可以容忍，那么 ξ_t 具有简单的闭式，且

因此，可以根据$t = 1$，……，c的递归关系，计算 $\phi_c(0)$的精确表达式

$$\phi_t(0) = \prod_{i=0}^{t} \lambda_i + (\mu_{t-1} + \lambda_t)\left[\phi_{t-1}(0) - \prod_{i=0}^{t-1} \lambda_i + \gamma_{t-1}\left(\prod_{i=0}^{t-2}(\gamma_i + \lambda_i) + \gamma_{t-3}\mu_{t-3}\prod_{i=0}^{t-4} \lambda_i\right)\right] \quad (4.6)$$

并且 MTTDL 表达式将变为精确值。否则，我们的计算将给出实际 MTTDL 数值的上界（知道为什么会这样吗？）。

这类闭式表达式的妙处在于，我们可以据此推断大量纠删码阵列的可靠性。例如，我们可以轻松计算出，125 个具有$m = 8$个数据设备的 RAID6 阵列（总共在 1250 个$1TB$磁盘设备上存储了$1PB$用户数据）的持久性数值为 3。这表明，随着存储数据规模扩展到 PB 级别，并行运行的传统 RAID 数据保护方案相当不可靠。如果我们将$m = 8$减少到$m = 7$，使同一存储系统中的用户数据存储总量达到 $875TB$，持久性数值便会增至 6。

4.2 案例研究：将广义模型应用于金字塔码

一个有趣的案例研究是，将上一节的广义马尔可夫模型应用于一种现代擦除码，例如 Microsoft Azure 存储的金字塔码 [39]。金字塔码旨在提升小规模设备故障时的恢复性能，并已应用于归档存储 [40]。金字塔码以标准最大距离可分码为基础构造，通过利用现有校验符号构造新的校验符号，在可恢复性、编码开销 和平均读取开销之间进行权衡；这些都是存储应用中需要优化的重要参数。我们以一个 (16,12) 最大距离可分码为基础，构造表 4.1 所示的一组 (18, 12) 金字塔码。这些可恢复性和读取开销数值的计算结果见文献 [39]。首先，我们注意到，可恢复性数值除以 100 后，便得到每种金字塔码对应的q_i值。指标平均读取开销表示访问每个数据块时额外设备读取次数的平均值。我们以 (18,12) 最大距离可分码发生一个块故障为例，说明如何计算该指标。如果故障块是冗余块（概率为 6/18），则可以直接访问数据块，因此平均读取开销为 1。否则，故障块是数据块（概率为 12/18），此时读取故障数据块需要十二次读取，其余十一个数据块各需一次读取。因此，平均读取开销为 (12+11)/12。总而言之，平均读取开销为$1 \times 6/18 + (12 + 11)/12 \times 12/18 \approx 1.61$。

事实上，对于通用的 (n,k) 最大距离可分码，当发生j次故障时，我们可以用以下广义表达式求出平均读取开销[4]

$$\Phi_j = \sum_{i=0}^{j} \frac{(ik + k - i)\binom{n-k}{j-i}\binom{k}{i}}{k\binom{n}{j}} \quad (4.7)$$

尽管平均开销并非影响修复过程的唯一指标，但为 r

[4]这个表达式很容易用归纳法证明，因此为节省篇幅，省略证明。

失效符号/块数量		0	1	2	3	4	5	6
通用 MDS 码	可恢复性（%）	100	100	100	100	100	100	100
	平均读取开销	1.0	1.61	2.22	2.83	3.44	4.06	4.67
基础金字塔码（BPC）	可恢复性（%）	100	100	100	100	100	94.12	59.32
	平均读取开销	1.0	1.28	1.56	1.99	2.59	3.29	3.83
广义金字塔码（GPC）	可恢复性（%）	100	100	100	100	100	94.19	76.44
	平均读取开销	1.0	1.28	1.56	1.99	2.59	3.29	4.12
不含全局符号的 GPC	可恢复性（%）	100	100	100	100	97.94	88.57	65.63
	平均读取开销	1.0	1.28	1.56	1.87	2.32	2.93	3.85

表 4.1：通用（18,12）最大距离可分码和若干金字塔码，以及 [39] 中给出的相关可恢复性/效率特征

了简化起见，我们假设修复率与该指标成反比。下面定义 e

$$\overline{\mu}_j \triangleq \delta\mu \ln\left((j+1)\Phi_{j+1}\right) \tag{4.8}$$

其中，μ 是每台设备的名义修复速率，δ 是用于建模相对带宽约束的常数（相对于最大距离可分码；对于最大距离可分码，该值归一化为 $\delta = 1$），用于根据平均读取开销指标反映修复率。该公式假设名义修复率与平均读取开销之间存在反指数关系。还需要更多现场数据来确认这种关系。

设当 i 台设备发生故障时，使用某个金字塔码所产生的平均读取开销为 χ_i。原论文并未推导出具有任意期望层级的给定金字塔码的 χ_i 闭式表达式，这是一个有趣的开放问题。我们将使用 [39] 中计算得到的 χ_i 来计算平均数据丢失时间。使用上一节的广义马尔可夫模型，我们进一步假设系统采用同质修复策略，即 $\mu_j = \overline{\mu}_{j+1}\ln((j)\chi_{j+1}))$。表 4.2 展示了部分结果，其中名义修复率为 $\mu = 1/168$（平均修复时间为 1 周）、$\eta = 10^{-3}$ 和 $\delta = 20$。我们观察到，基础金字塔码和广义金字塔码凭借高效的修复机制，能够提供更好的持久性指标。随着 λ 接近于零，修复频率降低，因此金字塔码的优势也会减弱。这一点可以从针对 $\lambda = 1/1.2M$ 给出的平均数据丢失时间结果中观察到。另一个有趣的观察是，在这些计算参数设置下，全局符号对于金字塔码维持期望的持久性水平至关重要。

5 可用性

可靠性的另一个重要指标是可用性，其定义为系统能够通过某种冗余方案提供数据访问的时间比例。可用性更多关注暂时无法访问的情况，即有足够数量的设备无法响应，导致数据无法重构。设备的寿命由可用和不可用的时段组成。设备正常运行时间 是指设备处于运行／在线状态的时间，而设备的停机时间 则是指设备无法响应／离线的时段。

故障/名义修复率		$\lambda = \frac{1}{200K}$	$\lambda = \frac{1}{500K}$	$\lambda = \frac{1}{1\text{-}2.M}$
通用 MDS 码	平均数据丢失时间（小时） 持久性（几个9）	2.2e+15 11	6.4e+17 13	1.3e+20 15
基础金字塔码（BPC）	平均数据丢失时间（小时） 持久性（几个9）	1.3e+17 13	5.2e+18 14	1.7e+20 16
广义金字塔码（GPC）	平均数据丢失时间（小时） 持久性（几个9）	1.32e+17 13	5.26e+18 14	1.76e+20 16
不含全局符号的 GPC	平均数据丢失时间（小时） 持久性（几个9）	1.83e+14 10	3e+15 11	4.1e+16 12

表 4.2：各种纠删码的平均数据丢失时间。

设备离线时，其中包含的数据会暂时不可用，且不会参与系统运行，除非设备重新上线。设备可能因各种原因而变得不可用。例如，存储设备、节点或网络交换机可能过载；操作系统可能崩溃或重启；控制器可能发生硬件错误；或者整个存储设备集群可能因维护而停机。绝大多数此类不可用事件都是暂时的，不会造成永久性数据丢失 [41]。

可用性最简单的定义之一，是预期正常运行时间除以预期正常运行时间与停机时间之和的比值，即

$$p_A \triangleq \frac{\mathbb{E}uptime}{\mathbb{E}downtime + \mathbb{E}uptime} \tag{5.1}$$

如果假设正常运行时间和停机时间均服从指数分布，则底层随机过程将成为马尔可夫过程，其状态图如图 5.1 所示。设设备的平均正常运行时间和平均停机时间分别为t_{up}和t_{down}。设备可用性为$p_A = t_{up}/(t_{up} + t_{down})$。此外，如果设备的平均故障时间（$1/\lambda$）[5]相对于平均正常运行时间和停机时间足够大（$\lambda t_{up} > \lambda t_{down} \gg 1$），我们

[5]这里我们隐含地假设设备寿命服从指数分布。事实上可以证明，当 $\lambda t_{停机} \ll 1$ [42] 时，情况确实如此。

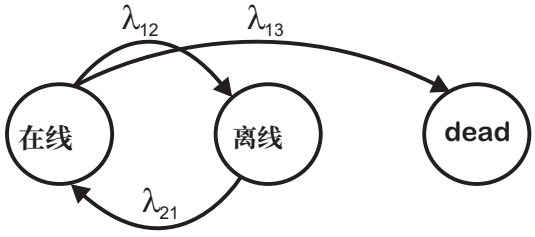


图 5.1：设备/节点可用性的马尔可夫链。

可以证明， [42], 连续马尔可夫模型的转移率为

$$\lambda_{12} = \frac{p_A - \lambda t_{up}}{t_{up} p_A},\ \lambda_{13} = \frac{\lambda}{p_A},\ \lambda_{21} = \frac{1}{t_{down}}. \tag{5.2}$$

注意，如果设备从未离线，即 $p_A = 1$ 或 $t_{down} = 0$，则有 $\lambda_{13} = \lambda = 1/t_{up}$。此外，利用引理 1 的结果，我们可以得出从**在线** 状态转移到**失效** 状态的概率为

$$p_{13} = \frac{\lambda_{13}}{\lambda_{12} + \lambda_{13}} = \frac{\lambda t_{up}}{p_A} = \lambda(t_{up} + t_{down}) \tag{5.3}$$

这将有助于下一节的讨论。

5.1 带超时的可用性

尽管状态图很清晰，但系统无法区分已失效的数据/校验设备与仅仅处于离线状态的设备。在这两种情况下，设备都没有响应，也无法知道它是否会重新上线。这一检测问题导致何时启动修复过程存在不确定性。决定何时触发修复的一种机制是，等待一段时间（称为超时），让离线/失效的副本恢复到在线状态。如果设备在超时前恢复在线，则无需修复；否则，系统会认为设备已经失效，并启动修复过程。超时时长的选择取决于系统希望以多积极的方式进行修复。必须针对特定系统优化超时时长，以避免低效地使用系统资源（例如不必要的修复）。

以下讨论见于 [42] ，由于其中提供了关于带超时的实际修复过程的有用信息，我们将在此重述。他们引入了系统参数 α ，规定超时时间为 αt_{down}。参数 α 使系统能够权衡对数据/奇偶校验潜在丢失的响应积极程度，以及发起不必要修复的次数。

令 Y_α 表示数据/奇偶校验生成后，离开在线状态且不再返回所经过的时间，即发生不可恢复错误的时刻，或超过超时期限所经历的时间。根据 [42],中的论述，可得以下引理。

引理 4 [42]: 随机变量 Y_α 的期望值为

$$\mathbb{E}Y_\alpha = \frac{(1 - p_{13})(1 - e^{-\alpha}) \left[t_{up} + t_{down} \left(1 - \frac{\alpha e^{-\alpha}}{1 - e^{-\alpha}} \right) \right]}{p_{13} + (1 - p_{13}) e^{-\alpha}} + t_{up} \tag{5.4}$$

证明：该引理的证明见 [42]。 □

选择合适 α 的一般做法是求解以下

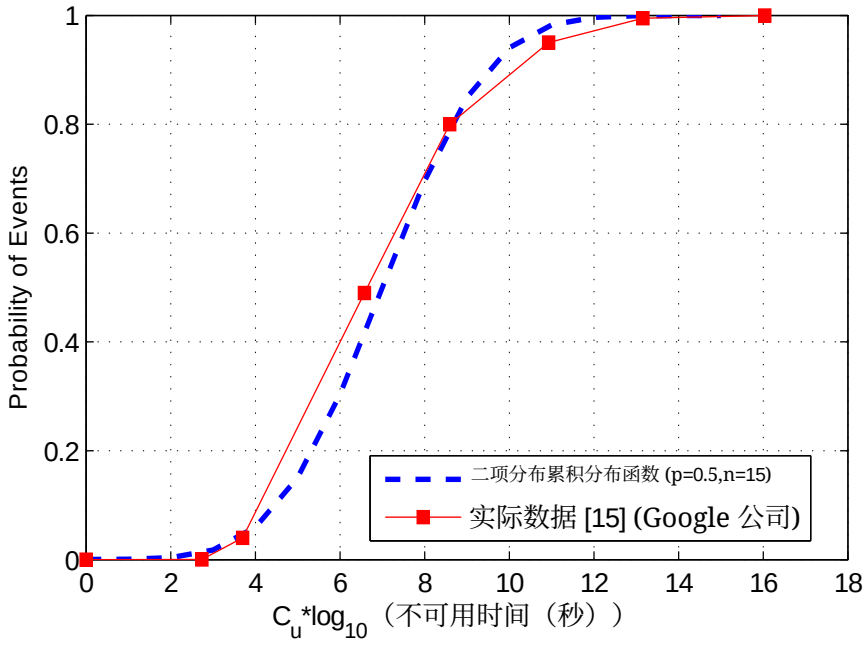


图 5.2：节点不可用时段持续时间的累积分布函数与二项分布拟合的对比。

方程：给定 λ、t_{up} 和 t_{down}

$$\frac{\lambda(1-p_{13})(1-e^{-\alpha})\left[t_{up}+t_{down}\left(1-\frac{\alpha e^{-\alpha}}{1-e^{-\alpha}}\right)\right]}{p_{13}+(1-p_{13})e^{-\alpha}}=1+\lambda(1-\alpha)t_{down}-p_{13} \tag{5.5}$$

在这里我们可以注意到，求解这个非线性方程无非是找出满足 $\mathbb{E}Y_\alpha+\alpha t_{down}=1/\lambda$ 的点，即预期超时时间与线性直线$y = 1/\lambda$ 的交点。这意味着修复过程将会启动，并有望在每个设备寿命周期内恰好生成一个新副本。同一项研究还给出了内存引入对整体修复过程影响的相关结果。显然，考虑哪些设备或节点需要修复，以及它们何时恢复在线状态，可以消除不必要的冗余，从而节省系统资源。不过，有趣的是，研究表明，当 α 经过精心选择时，带内存修复并不比不带内存修复好多少。

5.2 案例研究：谷歌不可用性数据

对数十个谷歌存储单元在一年期间收集的数据（每个存储单元包含 1000 到 7000 个存储节点） [41] 表明，不可用事件中持续时间超过 15 分钟的不到 10%。我们以秒为单位表示不可用事件持续时间，并使用$C_u \times \log_{10}$(“不可用事件持续时间”)来映射持续时间指标，其中 C_u

是一个合适的常数。这样的变换使我们能够恰当地将二项分布累积分布函数拟合到真实数据上，如图 5.2 所示。二项分布的参数为p = 0.5 和n = 15，以C_u = 3.7 拟合数据。尽管 [41], 中给出了节点不可用性的相关结果，我们预计设备/磁盘不可用性也会呈现类似趋势。

根据基于 Google 的数据进行的二项分布近似，t_{down}的平均停机时间约为 0.03 小时。[41] 还报告称，他们通常会等待 15 分钟（0.25 小时），然后才开始恢复不可用节点上的数据。这意味着 $\alpha = 8$。3。我们可以设定年故障率，即采用指数时间分布，λ 或设定正常运行时间 t_{up}。通常，测量t_{up}会更合理，并通过求解方程（5.5）计算出 λ。利用该数值和估算的修复率，可以按照前文所述进行平均数据丢失时间计算。

根据谷歌的论文，由于论文未报告标准正常运行时间测量值，因此可以方便地假设年故障率为 %4（据报告，在许多数据中心和现场测量中观察到的情况也表明这一数值更为现实 [41]），并计算t_{up}。基于这些假设，我们求解方程（5.5），得到t_{up}为 218,089 小时。这意味着 $\lambda_{13} = 0$。99584，接近于 1。当然，设备的不可用性会因环境以及与其他系统功能组件之间的相关因子而发生显著变化。真实数值会改变我们的计算结果，但肯定不会改变此处介绍的方法。

6 冷存储的多维马尔可夫模型

本节聚焦于一种特定的数据存储场景，该场景广泛应用于归档场景：数据只写入一次，却会被读取多次，而且读取频率很低。此外，针对这一应用，我们阐述了冷存储系统中的重要组件，其中可靠性和可用性都需要同时考量。因此，这是一个很好的应用案例，能够体现我们在前几章中关于可靠性和可用性所奠定的基础。本节内容主要涵盖于 [43]。

冷数据存储系统用于帮助机构长期保存数字归档。冷数据存储与温/热数据存储的共同点是，数据都存储在某种物理介质上，以便日后读取。不过，在冷存储中，写入和读取操作不一定在同一地理位置进行。因此，通常需要第三方协助，将介质与驱动器送到一起。另一方面，这种分解式系统的可靠性建模面临一些其他温/热存储方案未必存在的挑战，例如故障检测以及载体缺失，这些问题最终都会导致数据不可用。本文提出了一种广义非齐次马尔可夫模型，将载体老化纳入考量，以满足当今冷数据存储系统的需求；在这类系统中，数据经过编码后分散存储于多个节点，以实现长期数据保留。我们推导出了系统整体可用性的有用下界和上界。此外，我们利用收集到的现场数据估计威布尔分布的参数，从而在一个横向扩展环境示例中准确预测载体的寿命。

在对冷数据存储系统（如磁带）进行准确的可靠性建模时，应将介质故障、故障驱动器行为、故障检测以及数据分散（即数据分散到不同独立冷存储节点的程度）

都纳入考量[44]。特别是，在基于磁盘的系统中，可通过定期执行系统/数据检查操作来检测故障系统行为和数据丢失，这种操作称为擦洗 [60]。此类擦洗过程可以纳入可靠性模型 [45, 46, 48]。相比之下，发生载体故障时，数据并未丢失，但在维护团队更换或修复载体之前，数据将暂时不可用。因此，实际的可靠性模型应分别处理数据持久性和可用性，并能够恰当地结合这些重要的数据相关性能指标。

关于冷存储系统，过去有多项研究针对特定应用研究了备份系统，其中保存数据副本（复制）是实现持久性的主要手段 [53]。尽管这些研究关注了故障检测问题，但其模型并未针对真正的冷存储环境需求进行配置，即需要将介质从一个地点运送到另一个地点、机械组件的不利影响、载体不可用性以及与驱动器相关的硬错误。在 [54], 中，研究扩展到了4副本情形，此时备份系统由具有不同故障率和修复率的磁带与硬盘驱动器组成，即一个异构存储网络。由于磁带系统和硬盘驱动器的存储介质及内部机械结构不同，随着副本数量增加，所提出的模型很快变得复杂。因此，将其扩展到4副本以上似乎颇具挑战性，且同一研究路线中并未提出系统性的扩展方案。在 [55], 中，研究提出了一个二维马尔可夫过程，用于对基于磁盘的分布式存储系统中的显式错误和潜在错误进行建模，其中假设故障检测几乎不耗费时间。由于研究仅考虑基于磁盘的系统，因此并未进一步扩展以考虑载体的存在。

由于数据相关密度估计和载体随时间老化现象的复杂性，需要采用多维马尔可夫模型，并可将其实现为仿真平台，用于估计数据丢失时间和数据不可用时间的分布。通过仿真数据可以得出平均检测时间（MTTD）和平均数据不可用时间（MTTDU）等已知统计量，并针对不同的系统参数选择进行比较，也可与参数空间有限时已知的少数理论结果进行比较。

6.1 通用模型描述

所提出的可靠性模型结合了数据驱动的密度估计（一种已知的分布拟合）和连续时间马尔可夫过程，以准确估计系统数据丢失的密度，以及平均数据丢失时间（MTTDL）和平均数据不可用时间指标（MTTDU）等相关的一阶统计量。该模型包含两类状态：节点状态和系统状态，它们共同构成并由马尔可夫状态表征。尽管所提出的模型通常适用于任何类型的冷存储，我们在此专门考虑横向扩展磁带库存储库系统。

尽管我们将节点状态的数量视为任意值，但为简化起见，我们将以冷数据存储领域最常采用的三种不同节点状态为例。这些状态定义为给定节点的可用 (A)、故障 (F)和已检测 (D)，并自动生成整个系统对应的马尔可夫状态。我们想提醒读者，典型的连续时间马尔可夫过程广泛应用于由硬盘或固态设备阵列组成的温/热存储设备

（例如 [56] 中所示）。以往的工作主要关注通常具有两种节点状态的马尔可夫过程，即可用和故障。实际上，我们的处理方式使模型成为以可配置参数构建的、对以往所有马尔可夫模型的推广。尽管因此无法推导出闭式表达式，我们仍将采用近似方法得出性能指标的解析结果，例如均值统计量的下界和上界。

为了说明所提出模型的一种实例化形式，假设冷存储˜系统由一个（˜n，k）最大距离可分（MDS）擦除码提供保护，其中 ˜n 是码字（块），而k 是有效载荷长度。请注意，在此设定下，传统的复制（拷贝）系统对应于$k = 1$。量$r = k/\tilde{n} = k/n$˜称为码率。换言之，由于使用最大距离可分码，我们假设 ˜n和k分别是n和k的倍数。考虑到编码/解码复杂度，且不失一般性，本文将始终假设$n = \tilde{n}$和$k = k$，以阐明核心思想。

我们注意到，节点状态与马尔可夫状态并不相同；事实上，一个马尔可夫状态可以包含三种节点状态。我们可以将每个马尔可夫状态想象为由三个桶组成，分别统计处于 A、F 和 D 节点状态的节点数量。此外，我们假设节点类型完全相同，且以相同的速率发生故障 λ，即构成同质存储网络。这意味着，由于发生物理且不可逆的错误，无法读取存放在该节点磁带上的数据。此外，系统中还运行着另外三个进程：其中两个是并发且相同的数据与载体修复进程，第三个是并发且相同的故障检测进程。本研究假设，磁带上会运行错误检测进程，并在检测到错误时对其进行修复。此外，机器人等介质载体也可以修复，因为若它们不可用，所有数据操作都会停止。我们假设机器人与数据修复器进程以及故障检测进程均服从指数分布，其速率分别为 ϕ、 μ 和 θ。

所提出的连续时间马尔可夫过程的复杂度与系统状态总数密切相关，而系统状态总数取决于节点状态。对于$s > 1$，即节点状态数大于一[6]，假设有i 个可用节点，其中$k \leq i \leq n$（每个节点包含一个数据块——请记住，这是实现完美数据重构的必要条件），那么我们将有$s - 1$ 个节点状态，共享$n - i$个数据块。在这种情况下，分解总数为

$$\binom{n-i+s-2}{s-2} \tag{6.1}$$

例如，图 6.1 展示了当节点状态为$s = 3$（分别用 A、D 和 F 表示）时，所有系统状态（马尔可夫状态）与k和n的关系。但在另一种情况下，可以加入“等待服务：QS”状态，使节点状态数增至 4，即$s = 4$。计算马尔可夫状态总数（一般情况下s）的闭式表达式如下（包括

[6]在我们的表述中，我们假设节点状态 A 和 F 自然存在于任何可靠性模型中。对于本研究所提出模型的进一步扩展，可以纳入检测、参与和老化等不同过程。

完全失效状态）

$$N_s = \sum_{i=k}^{n} \binom{n-i+s-2}{s-2} + 1 = \sum_{i=0}^{n-k} \binom{i+s-2}{s-2} + 1$$
$$= \binom{n-k+s-1}{n-k} + 1 \tag{6.2}$$

请注意，对于一般情况，我们可以进一步将N_s表示如下

$$N_s = 1 + \binom{n-k+s-1}{n-k} \tag{6.3}$$
$$= 1 + \binom{n-k+1}{1} + \sum_{i=2}^{s-1} \binom{n-k+i-1}{i} \tag{6.4}$$
$$\geq \sum_{i=0}^{s-1} \binom{n-k+1}{i} \tag{6.5}$$

由此可清楚看出，方程（6.5）中的等式仅在$s = 2,3$时成立。对于较小的s，方程（6.5）可用作精确的近似。请注意，从（6.3）推导到（6.4）时，我们使用了归纳法。下文将给出N_s更紧的上界和下界，并渐近分析最终马尔可夫可靠性模型的复杂度。

注意，利用N_s表达式中的范德蒙德卷积，我们可以将其重写为$s > 1$

$$N_s = \binom{n-k+s-1}{s-1} + 1 \tag{6.6}$$
$$= \sum_{j=1}^{s-2} \binom{s-2}{j} \binom{n-k+1}{j+1} + n - k + 2 \tag{6.7}$$
$$\geq \sum_{j=2}^{s-1} \left(\frac{s-2}{j-1}\right)^{j-1} \binom{n-k+1}{j} + \sum_{j=0}^{1} \binom{n-k+1}{j} \tag{6.8}$$

由此我们可以推知

$$N_s \geq \sum_{j=0}^{s-1} \binom{n-k+1}{j}. \tag{6.9}$$

Note，式 (6.8) 中的下界比式 (6.9) 中的下界更紧。对于上 r

界，我们观察到

$$\begin{aligned} N_s &= \sum_{j=1}^{s-2}\binom{s-2}{j}\binom{n-k+1}{j+1}+n-k+2 \\ &= \sum_{j=2}^{s-1}\binom{s-2}{j-1}\binom{n-k+1}{j}+\sum_{j=0}^{1}\binom{n-k+1}{j} \\ &\le \sum_{j=0}^{s-1}\binom{s-1}{j}\binom{n-k+1}{j}=\sum_{j=0}^{s-1}\binom{s-1}{j}\binom{n-k+1}{n-k+1-j} \\ &= \binom{s+n-k}{s-1} \end{aligned} \tag{6.10}$$

其中，式（6.10）由式（6.10）利用帕斯卡三角不等式得到；对于任意正数$m \geq c$，该不等式为

$$\binom{m}{c}=\binom{m-1}{c-1}+\binom{m-1}{c} \tag{6.11}$$

由于 $\binom{s+n-k}{s-1}=\binom{s+n-k}{nk+1}$，我们可以推得：对于固定的$s \ll n$，若缩放$k$与某个较大的$n$呈线性关系，即$k=\alpha n$，则对于任意 $\{\alpha: 0<\alpha<1\}$，系统（马尔可夫）状态的总数将按$O(n^{\min\{s-1,n-k\}}) = O(n^{\min\{s-1,(1-\alpha)n\}}) = O(n^{s-1})$的尺度增长。换言之，除非码率趋近于 1（$r \to 1$），即$k$相对于$n$呈次线性增长，且常数$n-k<s$，否则我们的仿真框架（及相应马尔可夫链）的复杂度会随节点状态数呈指数增长。在这种情况下，系统状态总数将按$O(n^{n-k})$的尺度增长，且不受节点状态数s的影响。

6.2 特殊情况：三节点状态马尔可夫模型

如前所述，假设我们有三种节点状态：A、D 和 F。因此，下一次状态变化只能产生三种（一般来说为 s）目的状态。例如，对于给定的状态索引（iA，jD，zF），表 6.1 汇总了所有可能的目的状态状态索引。表中，DS 表示目标状态，并且 i、j、z应满足以下不等式

$$k \le i \le n,\ 0 \le j \le n-k,\ 0 \le z \le n-k,\ i+j+z=n$$

当前状态	目的状态	转移率
iA,jF,zD	$(i-1)$A,$(j+1)$F,zD	$i\lambda$
iA,jF,zD	iA,$(j-1)$F,$(z+1)$D	$j\theta$
iA,jF,zD	$(i+1)$A,jF,$(z-1)$D	$z\mu$

表 6.1：状态转移和速率

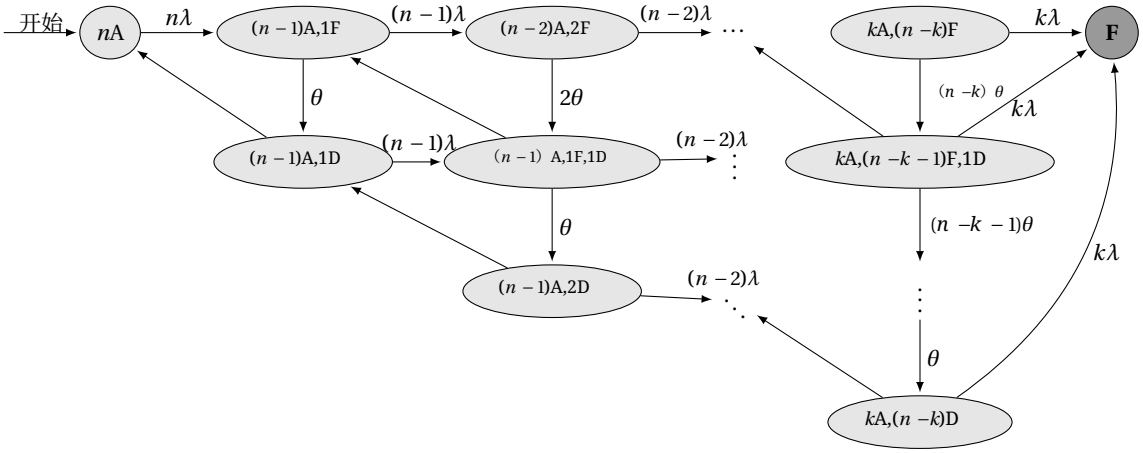


图 6.1：节点状态的广义马尔可夫模型，即由可用性（A）、检测（D）和故障（F）表示。状态 **F** 表示完全故障/不可用性。请注意，由于 $s = 3$，相应的马尔可夫模型可以表示在二维平面上。一般而言，马尔可夫模型是 $s - 1$ 维的。

仿真模型中未考虑的一项内容，是将三索引状态名称转换为单索引名称的方法；为方便仿真，该单索引名称的取值范围为 0 到$N_3 - 1$。一种简单直接的方法是令公共索引c_{ind} 为

$$\begin{aligned} c_{ind} &= \frac{1}{2}(j+z)(j+z+1)+z+1 \\ &= \frac{(n-i+1)!}{2(n-i-1)!}+z+1 = \binom{n-i+1}{2}+z+1 \end{aligned} \tag{6.12}$$

其中，将 $j + z = n - i$ 代入以得到第一个等式。请注意，c_{ind}与 (i , j , z) 之间存在一一对应关系，因此我们可以类似地求出索引 c_{ind}的逆变换。要从给定的 c_{ind}求出 (i , j , z)，首先需要找到最大的 $i \in \{k, k+1, \ldots, n\}$，使得 $c_{ind} > i(i+1)/2$。根据方程（6.12），由于已知 n 和 i，因此可以求出 z。最后，我们利用 $i + j + z = n$ 这一事实来确定 j。

此外，c_{ind} 的最大值应在状态索引（k,0, $n - k$）处取得。由于 $N_s =$

$\max\{c_{ind}\}+1$，可以证明以下结论成立

$$
\begin{aligned}
N_3 &= \binom{n-k+1}{2} + n-k+2 && (6.13)\\
&= \frac{1}{2}\left((n-k+1)(n-k)+2(n-k+2)\right) && (6.14)\\
&= \frac{1}{2}(n-k+2)(n-k+1)+1 && (6.15)\\
&= \sum_{i=1}^{n-k+1} i+1 = \sum_{i=0}^{n-k}(i+1)+1 && (6.16)
\end{aligned}
$$

请注意，当 $s=3$ 时，最后一个等式与公式（6.2）给出的结果相同。为了推导性能表达式并将一般模型应用于特定的实际应用，我们将在接下来的讨论中假设 $s=3$ 。

6.3 冷存储中的故障类型和载体不可用性

在温/热存储领域，硬错误场景已得到充分理解，即驱动器与存储介质紧密耦合的情况。此外，由于这种耦合，不存在单独的载体可用性问题。然而，在冷数据存储环境中，同时将硬错误和载体可用性纳入可靠性模型则要复杂得多。

让我们以磁带库系统为例进行说明。其中一个基本挑战是，机器人（载体设备）在发生故障之前只能执行一定次数的交换—交换（往返行程）。由于这一限制取决于时间和使用频率（系统负载），它会给当前的马尔可夫模型带来非齐次性。此外，同一系统中存在两种导致老化的驱动力：（1）用户数据访问模式，这在冷存储环境中通常影响较小；（2）由于系统/数据修复操作而在内部生成的访问请求，这些请求将导致额外的机器人交换、驱动器装载/卸载周期、磁带定位等操作，从而满足系统可靠性目标。对于其他常见的冷存储替代方案，也可以观察到类似现象。

为简化起见，我们限定了所考虑的因素，并假设有三个主要因素，其中两个直接影响数据持久性，另一个只改变数据的不可用性。其中一个因素是驱动器读取故障。在温/热存储中，这类硬错误也相当常见：由于驱动器内部的错误纠正解码失败而导致不可纠正错误，驱动器会因此无法读取数据。不可纠正错误率（UCER）通常以读取的字节数或比特数对应的错误数来表示，且通常由随机噪声效应引起。通常假设这种硬错误机制与时间无关；根据独立性假设，至少发生一次读取错误的概率可计算如下：

$$\epsilon = 1-(1-UCER)^{tape_capacity} \tag{6.17}$$

wh这里我们假设最坏情况，即批量读取，也就是读取整盘磁带，并且

磁带_容量是磁带可存储的字节数。请注意，在冷存储中，这种最坏情况相当常见。

另一种是存储介质（磁带）损坏。在我们的研究中，我们对磁带因制造原因（早期失效期）而在初始阶段发生损坏的概率，以及在其寿命后期因湿度、压力和严苛温度条件等外部因素而发生损坏的概率进行建模。我们假设这些因素是静态的，并且在故障检测和纠正后仍会持续存在于冷存储中所存储数据的整个生命周期内。基于这一考虑，我们使用 $\kappa > 0$ 来对这种损坏概率进行建模。更具体地说，我们认为，给定磁带库中的磁带损坏比例为 $\kappa \times 100\,\%$。

假设存储库机器人（载体）在发生故障并最终无法完成存储库发起的任务（包括检测与修复流程）之前，可以进行m 次交换（往返行程）。我们假设所有存储库中的机器人类型和质量都相同。根据从本地存储库系统中提取的可用数据，我们将说明m 可以建模为服从威布尔分布的随机变量，其形状参数（g）和尺度参数（y）取某些值。

通常假设交换之间的时间服从速率为 ω 的指数分布。速率 ω 取决于多个参数，例如使用该系统的用户数量、横向扩展环境中的库总数、一年中的时间等。因此，机器人失效时间（由随机变量 Y 表示）是 m 个速率均为 ω 的指数分布之和，即服从伽马分布，其概率密度函数如下：

$$f_Y(y;\omega,m) = \frac{\omega^m y^{m-1} e^{-\omega y}}{\Gamma(m)} = \frac{\omega^m y^{m-1} e^{-\omega y}}{(m-1)!} \quad \omega, m > 0. \tag{6.18}$$

其中， $\Gamma(.)$ 是完全伽马函数。请注意，由于伽马分布不具有无记忆性，因此需要通过引入随时间变化的条件累积分布函数（载体存活概率）来考虑老化，具体如下：

$$\begin{aligned} \beta(t) &= P(Y > t | l \text{ exchanges made}) \\ &= 1 - P(Y < t | l \text{ exchanges made}) \\ &= \frac{\Gamma(l,\omega t)}{\Gamma(l)} = \frac{\Gamma(l,\omega t)}{(l-1)!}, \quad t > 0. \end{aligned} \tag{6.19}$$

其中，假设机器人进行了 l $<m$ 次交换。在这种情况下，后一个条件概率也服从伽马分布，其概率密度函数为 $f_Y(y;\omega,l)$。此外，所包含的上不完全伽马函数定义如下：

$$\Gamma(l,\omega t) = \int_{\omega t}^{\infty} x^{l-1} e^{-x} dx \tag{6.20}$$

另一方面，硬错误率为 $\eta = 1-(1-\epsilon)(1-\kappa)$。请注意，我们需要修改模型以补偿硬错误。请注意，硬错误会将从当前状态（iA，jF，zD）到目标状态（（$i-1$）A，（$j+1$）F，zD）的状态转移拆分，而原本该状态转移以 $i\lambda$ 的速率发生。与以往研究中的观察结果类似，对于每个满足 i $>k$ 的状态，我们需要添加一条通往完全故障（**F**）状态的状态转移，以便纳入

当前状态	目的状态	转移率
iA, jF, zD	$(i-1)$A,$(j+1)$F,zD	$i\ \lambda\Delta_i$
iA, jF, zD	**F**	$i\lambda(1-\Delta_i)$
iA, jF, zD	iA,$(j-1)$F,$(z+1)$D	$j\theta$
iA, jF, zD	$(i+1)$A, jF, $(z-1)$D	$z\mu$

表 6.2：状态转移及相应的转移率

硬错误。对于拥有 i 个可用节点的情况，我们最多可以容忍 $i\ -k\ -1$ 个并发硬错误，从而成功到达状态（（$i-1$）A，（$j+1$）F，zD）。假设相互独立，该事件发生的概率为

$$\Delta_i := \sum_{l=0}^{i-k-1} \binom{i}{l} \eta^l (1-\eta)^{i-l} = 1 - I_\eta(i-k, k+1) \tag{6.21}$$

其中，l 表示重建期间或定期数据检查时同时发生的硬错误数量。此外，我们使用正则化贝塔函数$I_x(a,b) = B(x; a,b)/B(a,b)$，以避免二项分布累积分布函数的不稳定性和精度问题。这里，$B(x; a,b)$称为不完全贝塔函数，其定义为

$$B(x; a, b) = \int_0^x v^{a-1}(1-v)^{b-1}\, dv \tag{6.22}$$

其完全贝塔函数$B(a,b) = B(1; a,b) = \Gamma(a)\Gamma(b)\ \Gamma(a+b)$。注意，存在如下极限 $\lim_{\eta\to 1}\Delta_i = 0$，此时所提出的模型简化为从 ($n$A,0F,0D) 到 **F**的简单状态转移，其速率为$n\lambda$。由于假设停留时间服从指数分布，因此在该状态中的平均停留时间为 $1/n\lambda$，可将其视为系统持久性的下界。另一方面，当 κ 和 ϵ 较小时，上界可由图 6.1 所示系统的可靠性很好地近似。最后，我们在表 6.2 中总结了新的状态转移表，其索引满足

$$k+1 \le i \le n,\ \ 0 \le j \le n-k-1$$
$$0 \le z \le n-k-1,\ \ i+j+z = n.$$

请注意，如果我们重新绘制图 6.1 所示的整体马尔可夫系统，并将硬错误纳入其中，图示会显得更加复杂。为了得到这样的状态转移表，我们作出了一些假设，具体如下。

- 在典型的修复过程中，只会选取k 个磁带进行修复。对于局部可修复码， [58], 这个数量可以减少。或者，也可以请求读取超过k 个磁带，并仅使用最早完成的k 次读取，以提高性能。

- 我们假设硬错误彼此独立，即数据在同一个存储库节点中不能有多个片段（数据块）。

$$\mathbf{Q} = \begin{bmatrix} -n\lambda & n\lambda\Delta_n & 0 & 0 & \dots & 0 & n\lambda(1-\Delta_n) \\ 0 & -(\theta+(n-1)\lambda) & \theta & (n-1)\lambda\Delta_{n-1} & \dots & 0 & (n-1)\lambda(1-\Delta_{n-1}) \\ \mu & 0 & -(\mu+(n-1)\lambda) & 0 & \dots & 0 & (n-1)\lambda(1-\Delta_{n-1}) \\ 0 & 0 & 0 & 0 & \dots & 0 & (n-2)\lambda(1-\Delta_{n-2}) \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & & -(k\lambda+(n-k)\mu) & k\lambda \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}$$

6.4 转移率、概率矩阵与载体可用性

转移率矩阵（TRM）**Q**（如上所示）是一个$N_s \times N_s$实值矩阵，其元素$q_{ij} \geq 0$表示从状态i转移到状态j的速率。我们广义模型的转移率矩阵如下所示。请注意，我们已将完全失效状态（**F**）纳入矩阵，因此最后一行变为全零向量。我们注意到，对角线元素满足

$$q_{ii} = -\sum_{j \neq i} q_{ij} \Rightarrow \sum_{j} q_{ij} = 0 \quad \forall i \in \{0, 1, \dots, N_s - 1\} \tag{6.23}$$

这意味着矩阵的各行之和必须为零。

此外，定义**Q**，其各项定义为$q_{ij} := q_{ij}/|q_{ii}|$，对所有$i$和$j$均成立。于是，转移概率矩阵（TPM）为$\mathbf{P} = \mathbf{I} + \mathbf{Q}$。请注意，这是均匀化技术[7]的精确版本，该技术通过使用离散时间马尔可夫链近似这一过程，来计算有限状态连续时间马尔可夫链的瞬态解。在推导性能上界时，这一表述将发挥作用。

上一小节的分析未将载体的可用性纳入状态转移矩阵。值得注意的是，尽管节点故障（例如磁带故障）与机器人可用性相互独立，但节点修复和故障检测机制高度依赖载体（机器人）的可用性；也就是说，描述故障检测和节点修复的速率也必须随时间变化。随着时间推移，除非载体得到足够及时的更新，否则检测率和修复率都会下降。

当节点因各种原因发生故障时，故障检测过程会立即启动。同样，当检测到这些故障时，相关的修复过程也会立即开始。因此，对于给定的运行时间t，系统中的机器人年龄和质量并不相同（可能是由于发生了更换等情况）。这会导致存储节点受到不平等的对待，因此，我们的仿真设置必须跟踪机器人被更换的索引，以便对老化现象进行建模。

[7]在该均匀化技术中，q_{ii}被替换为 $\gamma \geq \max |q_{ii}|$。

为简单起见，假设每个节点只有一个载体（通过取平均值的论证，可以将其推广到多个载体，而不改变以下讨论），并令ψ_o（t）表示在时间t系统中有$o \in \{0, 1, \ldots, i\}$个可用机器人的概率（以特定的$i$个节点集合为条件），这些机器人的生存概率为$\beta_{s_1}$、$\beta_{s_2}$、…、$\beta_{s_i}$，其中$s_m \in \{1, 2, \ldots, n\}$。可以证明

$$\psi_o(t) = \Re\{F_{\mathscr{P}}(o)\} - \Re\{F_{\mathscr{P}}(o-1)\} \tag{6.24}$$

其中，R{.}表示实部，$F_{\mathscr{P}}$(.)是泊松二项分布的累积分布函数，定义如下：

$$F_{\mathscr{P}}(o) = \frac{1}{n+1} \sum_{l=0}^{i} e^{-\frac{2\sqrt{-1}\pi l o}{n+1}} \prod_{m=1}^{i} (1 + (e^{\frac{2\sqrt{-1}\pi l}{n+1}} - 1)\beta_{s_m}(t)) \tag{6.25}$$

其中，p–1是方程$x^2 = -1$的一个复数解。另一方面，由于本研究假设故障、检测和修复均服从指数分布，且检测和载体修复只能依次发生，因此，多个独立指数分布之和自然也服从指数分布，这并不令人意外。然而，为了使后续分析在解析上易于处理，本小节将采用一阶近似[8]。具体而言，我们假设速率分别为 ϕ、θ_1、……、θ_x的$x + 1$ 个指数分布之和近似服从速率为$R_{\boldsymbol{\theta}}(\phi)$的指数分布，其中

$$R_{\boldsymbol{\theta}}(\phi) = \frac{1}{\frac{1}{\phi} + \sum_{c=1}^{x} \frac{1}{\theta_c}} \tag{6.26}$$

其中 $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_x]$。当系统检测到故障事件时，状态会从起始状态 ($i$A, jF,zD) 转移到目标状态 (iA,($j-1$)F,($z+1$)D)，其中$j > 0$。执行检测时，我们需要j 个机器人才能完成这一过程。如果发现机器人数量不足，例如$b < j$个，那么我们需要修复$j - b$个机器人，使总数达到j个，以执行检测过程。假设在时刻t，已知有l 个故障机器人，满足$0 \le l \le j \le n - k$。那么，条件修复率，即马尔可夫模型中发生检测状态转移的速率，为 $(j - l)\theta + lR_{\boldsymbol{\theta}}(\phi)$。因此，对所有可能情况求和

[8]尽管在数值结果部分，我们将通过模拟实际分布来说明这一假设是一个很好的近似。

l，即可得到无条件节点故障检测率：

$$\theta_j(t;\phi) = \sum_{l=0}^{j}\big((j-l)\theta + lR_{\boldsymbol{\theta}}(\phi)\big)\psi_{j-l}(t) \tag{6.27}$$

$$= j\theta - (\theta - R_{\boldsymbol{\theta}}(\phi))\sum_{l=0}^{j} l\psi_{j-l}(t) \tag{6.28}$$

$$= j\theta - (\theta - R_{\boldsymbol{\theta}}(\phi))\sum_{m=1}^{j}(1-\beta_{s_m}(t)) \tag{6.29}$$

$$= j\theta - \frac{\theta^2}{\theta+\phi}\sum_{m=1}^{j}(1-\beta_{s_m}(t)) \tag{6.30}$$

其中 $s_m \in \{1,\ldots,n\}$。注意，对于任意 t，我们都有如下不等式，

$$j\theta - \frac{j\theta^2}{\theta+\phi} \le \theta_j(t;\phi) \le j\theta \tag{6.31}$$

这意味着，当 $\phi \to \infty$，即机器人修复瞬时完成时，检测速率将为 $j\theta$，这与图 6.1 中给出的无机器人故障情形相同。

6.4.2 时变载体修复

检测到节点故障后，系统会立即开始修复过程；修复过程完成意味着状态从起始状态（iA，jF，zD）转移到目标状态（（$i+1$）A，jF，（$z-1$）D），适用于所有满足 $z>0$ 的起始状态。

假设在时刻 t，系统处于状态（iA，jF，zD），且 i 个可用节点中有 l 个节点的载体机器人已经故障。请注意，对于经典 MDS 码，需要有k 个辅助节点，才能成功完成数据请求[9]。进一步假设这些请求中有 x 个来自故障集合，而x 个来自可用且正常运行的节点，其中 $k-$。由于采用无放回抽样，发生这种情况的概率由超几何分布给出[10]。在这种情况下，我们必须先等待 x 个故障载体完成修复，这段等待时间由最大修复时间决定，通常不服从指数分布。事实上，可以证明，该分布等于若干指数分布之和；本小节假设该分布接近另一个速率为 $1/\phi\sum_{m=1}^{x}1/m$的指数分布，其中速率中的调和和可近似为

$$hs(x) := \sum_{m=1}^{x}\frac{1}{m} \approx \log(x) + \zeta + \frac{1}{2x} - \frac{1}{12x^2} + \frac{1}{120x^4} \tag{6.32}$$

其中 $\zeta = 0.5772156649$ 被称为欧拉–马歇罗尼常数。

[9]存在各种网络编码，其完全恢复 [59]可能需要访问多于或少于k个辅助节点，并部分访问节点内容。如果改用这类编码，本文的讨论只需稍作调整。[10]有放回抽样则会得到二项分布统计量。

任何一个z已检测到的节点收集到所有必要的修复信息后，都会开始执行修复过程所需的计算，并将修复后的数据写入相应的存储单元。但写入过程至少需要一个可用的运载器/机器人。其可用性分析与检测过程的相同情况（每个节点使用自己的机器人检测故障）相当类似，因此此类情况发生的速率表示为 $\mu_z(t;\phi)$，其表达式为

$$\mu_z(t;\phi) = z\mu - \frac{\mu^2}{\mu+\phi}\sum_{m=1}^{j}(1-\beta_{s_m}(t)) \tag{6.33}$$

另一方面，条件修复率（以x和l为条件）可表示为

$$\mu_{iz}(t;\phi,k|x,l) = \frac{\binom{i-l}{k-x}\binom{l}{x}}{\binom{i}{k}}\left(\frac{1}{\mu_z(t;\phi)} + \frac{hs(x)}{\phi}\right)^{-1} \tag{6.34}$$

其中$s_m \in \{1,\ldots,n\}$。最后，无条件修复率可通过对所有x和l求和得到，如下所示，

$$\mu_{iz}(t;\phi,k) = \sum_{l=0}^{i}\psi_{i-l}(t)\sum_{x=0}^{l}\mu_{iz}(t;\phi,k|x,l) \tag{6.35}$$

$$= \sum_{l=0}^{i}\psi_{i-l}(t)\mu_z(t;\phi) \tag{6.36}$$

$$+ \sum_{l=0}^{i}\psi_{i-l}(t)\sum_{x=1}^{l}\mu_{iz}(t:\phi,k|x,l) \tag{6.37}$$

注意，如果 $\phi\to\infty$，即假设机器人立即进行修复，则有

$$\lim_{\phi\to\infty}\mu_{iz}(t;\phi,k) = \sum_{l=0}^{i}\psi_{i-l}(t)\sum_{x=0}^{l}\frac{\binom{i-l}{k-x}\binom{l}{x}}{\binom{i}{k}}z\mu = z\mu \tag{6.38}$$

这意味着，当机器人修复是瞬时完成时，节点修复率将为$z\mu$，这与没有任何机器人故障时相同。最后，我们在表 6.3 中总结了新的状态转移表，其索引满足以下不等式

$$k \le i \le n, \tag{6.39}$$

$$0 \le j \le n-k, \tag{6.40}$$

$$0 \le z \le n-k, \tag{6.41}$$

$$i+j+z = n.$$

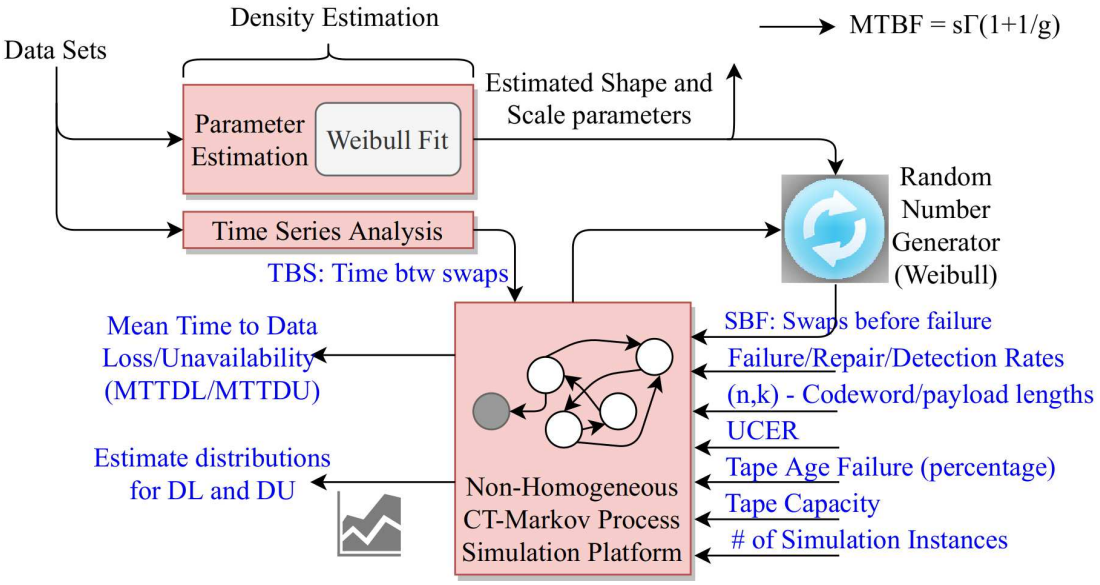


图 6.2：在 MATLAB 中实现的数据启发式整体建模与仿真平台。通过假设机器人交换性能数据服从威布尔分布，将密度估计问题简化为参数估计。在此图中，s 表示尺度参数，g 表示形状参数。MSBF：平均故障前时间。

6.5 性能上下界

对于给定的有限载体修复率 $\phi < \infty$，如果我们让交换率趋于很大，载体维修将无法跟上，最终导致载体完全不可用。在这种情况下，可以用闭式求出性能下界。我们注意到，在载体完全不可用的情况下，不会进行故障检测，因此在冷存储场景中也就不会进行数据修复；于是，生存时间取决于系统所处的状态，以及硬错误是否会导致不可恢复状态转换。根据这一观察，可以推导出下界（LB）。

我们先将最终未进入完全 1

当前状态	目标状态	转移率
i A,j F,zD	$(i-1)$A,$(j+1)$F,zD	$i\ \lambda\Delta_i$
i A,j F,zD	**F**	斜体 $\lambda(1-\Delta_i)$
i A,j F,zD	iA,$(j-1)$F,$(z+1)$D	$\theta_j(t;\phi)$
i A,j F,zD	$(i+1)$A,jF,$(z-1)$D	$\mu_i z(t;\phi,k)$

表 6.3：状态转移及其对应的转移率。

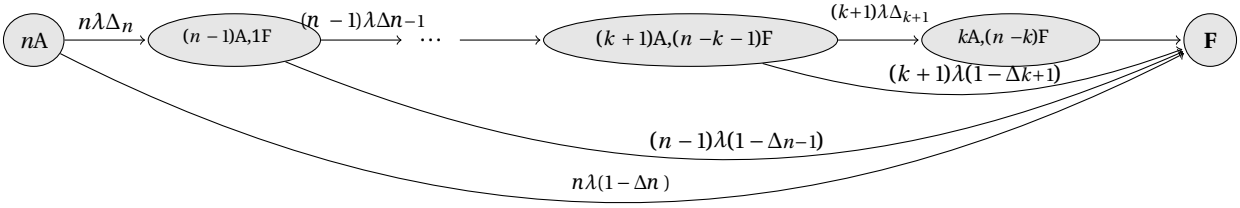


图 6.3：一般的连续时间马尔可夫模型可简化为具有以下转移率的单维模型。

数据丢失（故障）状态的条件概率改写为显式积分形式，其中有i个可用节点， Δ_i

$$\begin{aligned}\Delta_i &= \frac{\Gamma(i+1)}{\Gamma(i-k)\Gamma(k+1)}\int_{\eta(t)}^{1} v^{a-1}(1-v)^{b-1}dv \\ &= \frac{i!}{(i-k-1)!k!}\int_{\eta}^{1} v^{i-k-1}(1-v)^{k}dv \\ &= (i-k)\binom{i}{k}\int_{\epsilon+\kappa-\epsilon\kappa}^{1} v^{i-k-1}(1-v)^{k}dv \end{aligned} \tag{6.42}$$

其中， $\eta = 1-(1-\epsilon)(1-\kappa)$ 为硬错误率。为推导下界，我们考虑载体完全不可用的情况。因此，不会进行故障检测，也就不会启动数据修复过程。这将得到图 6.3 所示的简化版马尔可夫模型。注意，当有i个可用节点时，导致完全故障的硬错误发生的概率为 $(1-\Delta_i)\prod^{n}_{j=i+1}\Delta_j$。故障发生前的总时间，是在每个访问过的系统状态中所花费的平均时间之和，即 $1/n\lambda$、$1/(n-1)\lambda$、...、$1/i\,\lambda$。最后，对所有可能的i求和，即可估算故障发生前所花费总平均时间的下界：

$$\begin{aligned} LB &= \sum_{i=k}^{n}(1-\Delta_i)\prod_{j=i+1}^{n}\Delta_j\sum_{j=i}^{n}\frac{1}{j\lambda} \\ &= \sum_{i=k}^{n}(1-\Delta_i)\prod_{j=i+1}^{n}\Delta_j\,(hs(n)-hs(i-1)) \\ &\approx \sum_{i=k}^{n}(1-\Delta_i)\prod_{j=i+1}^{n}\Delta_j\left(\log\left(\frac{n}{i-1}\right)^{1/\lambda}+\frac{1}{\lambda}\sum_{l=0}^{1}(-1)^{l}\left(\frac{1}{2n_l}-\frac{1}{12n_l^2}+\frac{1}{120n_l^4}\right)\right) \end{aligned} \tag{6.43}$$

其中 $\Delta_k = 0$和 $n_l = n - l(n-i+1)$。注意，如果没有硬错误，即 $\Delta_i = 1$对于 $i = n,\ldots,k+1$，我们就可以得到简化的下界，其闭式表达式为 $\sum^{n}_{j=k}1/j\,\lambda$。一般情况下，我们有

$$LB \le \sum_{j=k}^{n}\frac{1}{j\lambda} = hs(n)-hs(k-1) \tag{6.44}$$

注意，近似是由式 (6.32) 得出的。

另一方面，如果我们令交换率趋近于零，就不再需要进行载体维修，从而实现载体的完全可用。在这种情况下，我们希望以闭式推导性能的上界。我们注意到，模型中只有一个吸收状态（故障状态），因此 TPM 已经处于标准形式，

$$\mathbf{P}_{N_s\times N_s}=\left[\begin{array}{c:c} \mathbf{L}_{N_s-1\times N_s-1} & \mathbf{R}_{N_s-1\times 1} \\ \hdashline 0\quad\dots\quad 0 & 1 \end{array}\right]$$

对于吸收马尔可夫链，我们知道，$\mathbf{I}-\mathbf{L}$ 矩阵的逆称为基本矩阵（记作$\mathbf{M}$），可表示为

$$\mathbf{M}=(\mathbf{I}-\mathbf{L})^{-1}=\mathbf{I}+\sum_{i=1}^{\infty}\mathbf{L}^{i} \tag{6.45}$$

其中，m_{ij} 项表示：当马尔可夫过程以暂态s_i为初始状态时，访问暂态s_j的期望次数。由于我们最初假设所有节点在开始时均可用，因此我们关注m_{1j}s，即假设系统在运行开始时处于状态nA。对于s_j，所有离开该状态的状态转移均服从指数分布，停留时间由其中的最小值决定，其同样服从指数分布，速率为 $-q_{jj}$。这意味着，每次访问s_j时的平均停留时间为 $-1/q_{jj}$。最后，上界可近似表示为

$$UB\approx-\sum_{j=1}^{N_s-1}\frac{m_{1j}}{q_{jj}} \tag{6.46}$$

请注意，这只是一种近似，因为 TPM 是对连续时间马尔可夫模型的近似。此外，我们只需考虑表 6.3 中给出的速率矩阵，而非表 6.2，即可通过解析方法评估包含机器人故障与维修过程的时变性能上界。这是可行的，因为我们将分布近似为指数分布，以保持马尔可夫性。稍后将在数值结果部分验证，对于所关注的参数范围，这种近似具有足够的准确性。

6.6 数据辅助建模框架

随着存储设备的内部工程细节日益复杂，以数据为中心的可靠性分析越来越受到关注。本节介绍一项针对冷存储/磁带存储场景的此类研究实例。为此，我们采用数据启发式方法，估算载体在发生关键故障之前的往返行程次数（在本文语境中即交换次数）。由于各种原因，当机器人无法继续在存储库系统中运行，直至被新机器人替换时，便会发生关键故障。在我们的磁带应用中，假设故障前机器人交换次数（SBF）服从威布尔分布；这一假设将通过使用企业级 Quantum 磁带库收集的现场数据加以验证。威布尔分布完全由两个相互独立的参数决定，即形状参数 (<style id='1'>g</style>) 和尺度参数 (<style id='3'>y</style>)。

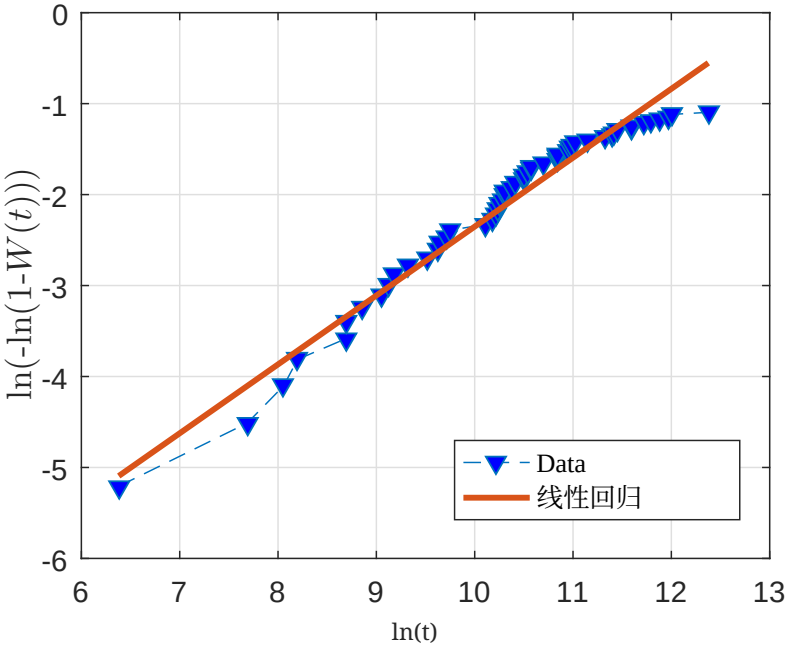
选择威布尔分布有两个原因。首先，它是文献中最常用的指数分布（单参数）的推广。换言之，通过选择适当的参数值，威布尔分布可以变换为指数分布。其次，它具有重尾特性，能够很好地刻画观测到的现场数据。我们发现，随着分布参数数量的增加，重尾分布能够准确地刻画各种类型的数据。例如，多项研究指出，数据对象大小往往呈现重尾分布，例如帕累托分布 [49]。另一方面，若干研究表明，重尾分布也能很好地刻画本地文件系统动态特性和文件大小 [50]，归档数据 [51] ，以及通过万维网存储和传输的数据 [52]。

我们注意到，基于可用的现场数据和威布尔假设，具有挑战性的密度估计问题被转换为参数估计问题。更准确地说，我们通过简单线性回归估计分布的形状参数（g）和尺度参数（y）。其次，我们获得交换/置换间隔时间分布的估计。利用同一数据集可以观察到，该分布具有指数尾部，因此只需估计一个参数（速率）。指数假设也相当巧妙，因为相应的计数过程将转化为在解析上易于处理的泊松分布。由于所估计的参数是系统利用率的函数，因而具有时间依赖性，我们将在仿真中测试一系列数值，以展示整体情况。建模框架的概要如图 6.2 所示。在该框架中，基于数据的参数估计值（$\hat{}g$ 和 $\hat{}y$）作为估计输入被馈入所提出的非齐次马尔可夫过程中。除这些输入外，我们还根据现场数据以及对 6TB磁带的使用经验，将其余仿真参数 λ、μ、θ、n、k、κ、ϵ 以及仿真实例数量设置为适当的数值。系统由（n，k）最大距离可分码保护。随机数生成器根据估计的威布尔分布选择一个随机的 SBF值，并重复这一过程，在仿真的每次迭代中使用一个独特的实现。我们通常进行 10000 次以上的仿真，以获得可靠的数值。

仿真平台的主要目的是估算总体数据丢失和/或不可用性的分布（即哪一种情况会先导致性能下降），同时展示这两个重要性能指标之间的隐含关系。换言之，我们最终可以相当有把握地对平均数据丢失时间和数据平均不可用时间指标进行数值估算。此外，交换次数的平均值为 $\hat{}y\Gamma(1+1/\hat{}g)$，该值将作为数值结果部分中机器人性能的指导依据。

请注意，威布尔分布参数（即g和y）的估计方法不止一种。本研究采用简单线性回归来估计威布尔分布的这些参数。不过，需要进行少量代数变换，才能将威布尔分布的累积分布函数（CDF）整理成适当的形式。因此，我们回顾一下由下式给出的威布尔累积分布函数$W(t)$。

$$W(t) = 1 - e^{(t/y)^g} \tag{6.47}$$



Figu图 6.4：机器人交换数据以及通过线性回归进行的威布尔参数估计 n.

该式可以重新整理并表示为以下线性方程

$$\ln\left(-\ln\left(1-W(t)\right)\right) = g\ln(t/y) = g\ln(t) - g\ln(y) \tag{6.48}$$

如果将纵坐标设为左侧表达式 $\ln\left(-\ln\left(1-W(t)\right)\right)$，横坐标设为 $\ln(t)$，并进行线性回归，就会得到一个线性函数，该函数自然具有截距($\mathscr{I}$)和斜率($\mathscr{S}$)。利用这些估计值，我们可以得出形状参数 ($\hat{}g$) 和尺度参数 ($\hat{}y$) 的估计值，如下所示，

$$\hat{g} = \mathscr{S}, \quad \hat{y} = \exp\left(-\mathscr{I}/\hat{g}\right). \tag{6.49}$$

换言之，直线的斜率应为形状参数，而尺度参数则需根据形状参数的估计值，按照公式 (6.49) 计算。为验证威布尔假设的准确性，我们记录了约 40000 次机器人交换，直至机器人停止运行。这些同等质量的机器人运行于 Quantum Scalar i6K 企业级磁带库中，该磁带库最多可容纳 12000 个磁带盒，并针对高密度进行了优化。根据公式 (6.48) 给出的表达式，我们将这些数据绘制在图 6.4 中，据此可以轻松求出截距和斜率，并用于计算形状参数 $\hat{}g = 0.76$ 和尺度参数 $\hat{}y = 491669$。对于满足 $9 \le \ln(t) \le 12$ 的 t 值，数据的累积是因为大多数机器人的寿命在该范围内呈对数分布。根据威布尔分布的估计参数，可以计算出发生严重机器人故障前的平均交换次数为 $\hat{}y\Gamma(1+1/\hat{}g) = 580747$ 次交换

表 6.4：仿真参数

参数	数值
λ （小时）	1/50000
μ （小时）	1/24
θ （小时）	1/8760
(n, k)	变量
ϵ （不可纠正错误率）	10^{-19}
t ape_c ap aci t y	6TB
$\hat{}g$ （形状参数，威布尔）	0.37 和 0.67
$\hat{}y$ （尺度参数，威布尔）	525985
κ	0.001
仿真次数	>10000

在数值结果一节中，我们将选择与这些数值大致相当的参数，以使结果和结论符合实际。最后需要指出的是，采用参数更多的分布可能会更好地拟合数据，但这只会带来极其有限的准确度提升，却要付出估计复杂度增加的代价。

6.7 数值结果

与冷存储（以及归档存储）平台通常的情况一样，我们主要关注读回性能，换言之，即数据检索性能。下面给出所提出的仿真与建模平台的一些数值结果。我们的目的是在同一张图中，以纵坐标表示可靠性（以平均数据丢失时间 MTTDL 衡量）和不可用性（以平均数据不可用时间 MTTDU 衡量），并展示它们随其他仿真参数的变化。横坐标可以是系统的某个参数，包括交换速率和载体修复率。由于仿真参数众多，且难以用二维图形展示或绘制高维数据，因此我们绘制二维图，并固定大多数仿真参数，仅改变交换/调换（*xph*）和载体修复率（ϕ）。前者通常随系统利用率变化，而后者由系统维护团队控制。选择改变这些参数的另一个原因是，它们会直接影响系统的不可用性：也就是说，在载体缺失（载体故障）期间，整体数据访问时间会增加，直至载体修复完成。特别需要指出的是，大多数载体设备（例如机器人）出厂时都会标明可确保可靠运行的最大交换/调换速率（例如每小时 840 次交换（*xph*） [47]）。我们将横坐标从较低的交换次数变化到高于标称最大值的范围，并以双对数图呈现结果。同样，在绘制以 ϕ 为参数的 MTTDU 时，我们固定交换/调换速率，并改变载体修复率，以观察修复频率对不可用性性能的影响。请注意，在给定时刻 *t*，系统可能处于这些性能曲线上的任意位置。

仿真的参数简要汇总于表 6.4 中。可以看到，我们

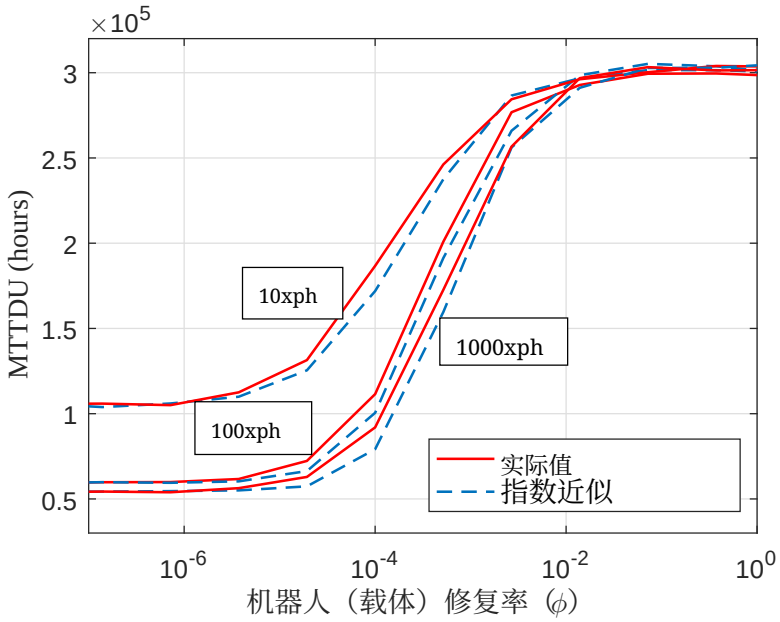


图 6.5：使用 (4,2) 最大距离可分码时，指数尾近似相对于平均数据不可用时间的准确度，随机器人修复率（ϕ）的变化；两种不同交换率分别为 10xph 和 100xph。

假设数据/磁带平均修复时间为一天，故障检测平均时间为一年，并以此作为起点。这些数值同样取决于具体应用，可根据使用场景进行调整。我们选取了几个示例半速率码参数，例如 (4,2) 和 (6,3)，并为其设定了不同的可靠性保证。横向扩展系统由n 个相同的磁带库组成，每个磁带库在收到请求时仅存储和处理一个数据块。每个磁带库都有各自专属的机器人，并包含多个存储单元，例如磁带。我们假设磁带和机器人的质量与类型均相同。横向扩展存储库系统的内部细节（例如磁带库总数M、每个磁带库中的磁带数量以及磁带架位置的几何布局等）的直接影响，都由威布尔参数估计（形状参数和尺度参数）及数据辅助建模框架纳入考量。这种数据分析使我们无需深入了解存储库系统的内部复杂性，并为我们提供每个磁带库交换次数的统计特性。随后，这些特性将作为前一节介绍的广义马尔可夫模型的输入。最后请注意，假设不可纠正错误率为 10^{-19}，远低于已知磁盘驱动器系统的不可纠正错误率 10^{-15}。这是由于新一代磁带技术 [57]能够提供很高的数据持久性保证。

我们的首次仿真结果见图 6.5，其中清楚地验证了此前提出的指数尾假设（例如表达式 (6.26)）适用于近似所提出的马尔可夫模型各个阶段中出现的非指数分布。我们采用了 (4,2) 最大距离可分码和三种交换率，即 10xph、100xph 和 1000xph，并以机器人修复率为函数，绘制了使用实际分布和指数尾近似（本次仿真中采用）得到的 MTTDU 结果。

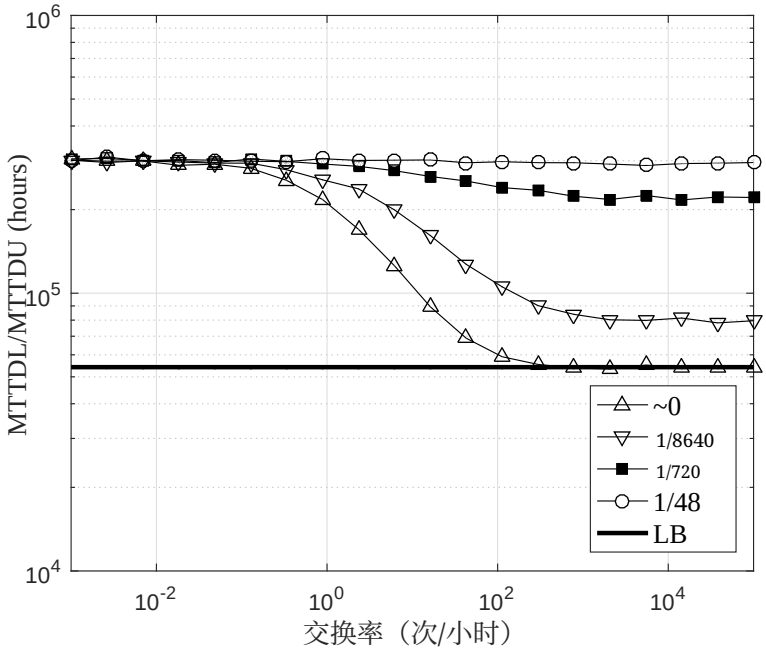


图6.6：对于形状参数为 0.67 的 (4,2) 最大距离可分码，MTTDL/MTTDU（小时）随交换率（每小时往返或交换次数）的变化情况，展示了不同修复率 ϕ。LB 表示推导出的下界。

初步观察之一是，我们的指数尾假设为实际 MTTDU 数值提供了一个下界。此外，实际值与近似值之间最坏情况下的差异不会影响几个*9*的数量；在业界，这一指标通常用于结合 MTTDL 性能指标来表示系统可靠性。在本小节余下部分中，鉴于指数尾近似的表达更简洁且便于分析，我们将使用该近似来呈现结果。

图 6.6 展示了 (4,2) 最大距离可分码的 MTTDL/MTTDU（小时）随交换率的变化情况。根据我们的数据观察结果，以及此前为估算尺度参数和形状参数而推导出的方程 (6.48) 和 (6.49)，我们发现最常见的形状参数为 0.37 和 0.67，尺度参数为 525985；相应地，在发生机器人故障之前，平均交换总次数分别为 695563 和 2200634。请注意，如果机器人仅能使用一年，这些数字则分别表示平均交换率为 79.4 次/小时和 251.2 次/小时。我们还在图中加入了可用性下界（由方程 (6.44) 给出），该下界不会随交换率的增加而变化。

如图6.6所示，随着交换率趋近于零（沿横坐标从右向左移动），可靠性逐渐趋近于图 6.1 中所介绍的系统模型的持久性；在该模型中，可用性问题不再对数据访问构成风险。另一方面，随着交换率增大（沿横坐标从左向右移动），平均数据不可用时间趋近于持久性下界。根据存储库系统的运行点，我们的模型清楚地展示了：如果机器人修复不够频繁，不可用性会如何随交换

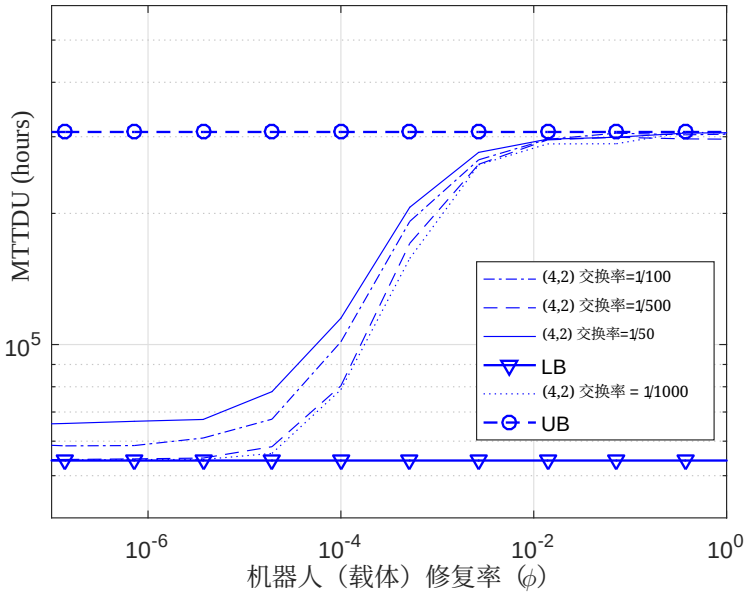


图6.7：(4,2) 最大距离可分码的 MTTDU（小时）随机器人修复率（ϕ）变化的关系，以及展示的几个交换率（exr）示例。LB 表示下界。

率变化。此外，从相同的仿真数据中可以观察到，机器人修复率为 $\phi = 1/48$ 时，系统维护的频率似乎已足够，因此在这一特定修复率下，可用性并未显著降低。

另一方面，从图6.6（以及图6.7和图6.8）可以看出，随着机器人修复率 ϕ 提高，也就是说，机器人修复得更加频繁，可用性也会提高。然而，提高到某个程度后，继续增大 ϕ 带来的改善就不大了；也就是说，系统中的机器人修复得足够快，不可用性不会导致显著的性能损失。为了找出最优机器人修复率，我们还需要绘制不同交换/互换率下 MTTDU 随 ϕ 变化的曲线。图6.7和图6.8分别展示了半速率最大距离可分码 (4,2) 和 (6,3) 的性能曲线。图中还包含使用方程 (6.44) 和 (6.46) 计算出的相应下界和上界。这两幅图有两个共同的有趣发现。其一是，超过一定交换率后，MTTDU 性能会趋于一致。例如，在图6.7中，$\phi = 1/500$ 和 $\phi = 1/1000$ 所对应的 MTTDU 性能差异并不显著。(6,3) 最大距离可分码在图6.8中也呈现出完全相同的趋势。因此，由于降低交换率可以提高可用性，我们可以确定一个最优修复率；超过该速率后，对于所有关注的交换率，都不会出现不可用性。从用户满意度和节能的角度来看，确定给定系统的最优交换率十分重要。关于这些图的第二个发现是，在码率固定的情况下，随着最大距离可分码的码块长度增大，相应的下界会变差。这是因为校验位的数量增长不足以弥补码块长度的增加。

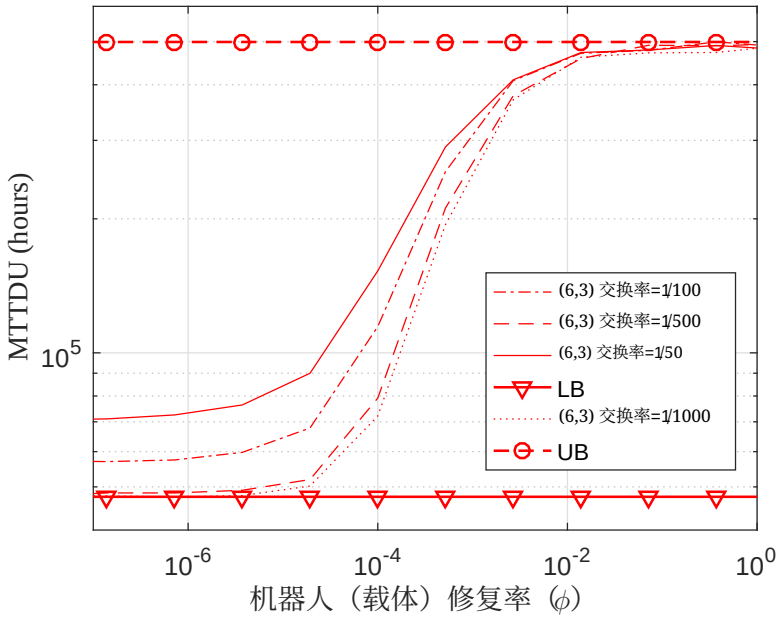


图 6.8：(6,3) 最大距离可分码的平均数据不可用时间（小时）随机器人修复率（ϕ）的变化，以及展示的几个交换率（exr）示例。LB 表示下界。

然而，利用推导下界时得到的表达式（方程 (6.44)），可以立即看出，当 n 趋于无穷大时，同一码率的不同规模最大距离可分码之间的性能差异将消失。根据图6.7和图6.8，我们可以分别量化这两种半速率码，即 (4,2) 和 (6,3) 最大距离可分码之间的差异。

7 高级建模

在我们之前的讨论中，可以认为，与这些模型相关的一系列扩展和马尔可夫分析，反映了对实际存储系统过于简单化的看法；尤其是此类系统的规模和尺度仍在不断增长。尽管之前的讨论是基于真实数据对现有解析方法的显著改进（因此非常适合用于快速粗略比较），但由于模型采用了恒定故障率和修复率等假设，仍然可能受到批评。然而，不幸的是，在可靠性模型中引入时间依赖性会增加复杂度，并可能妨碍解析可靠性估计。此外，我们观察到，将规范模型扩展到多磁盘容错系统时，由于底层指数故障／修复时间分布具有无记忆性，在重建建模方面会遇到问题。

7.1 纳入扇区错误

此前提出的将硬错误纳入马尔可夫模型的方法，难以准确表示潜在扇区错误；研究发现，这类错误比不可恢复的硬位读取错误更常见 [61]。文献中对潜在扇区故障／清理有两种建模方法：一种是将其视为磁盘故障，另一种是将其建模为独立的随机过程。本文采用后一种方法，基于定期擦洗来消除潜在扇区错误。对于给定扇区，设定固定的擦洗周期T_S、该扇区的负载l、由写操作导致扇区错误的概率P_w，以及写请求占系统总负载的比例r_w，则在任意时刻t（与时间无关），给定扇区发生不可恢复错误的概率为 [62]

$$P_S = \left(1 - \frac{1 - e^{-lT_S}}{lT_S}\right) P_w r_w \tag{7.1}$$

在同一项研究中，推导出了一个如下形式的时间相关概率

$$P_S(t) = \left(1 - e^{-l(t \mod T_S)}\right) P_e \tag{7.2}$$

其中P_e是单扇区故障概率。遗憾的是，将负载项l纳入此类计算，以得出恰当的扇区故障模型，仍然是一个开放问题。P_S可用于计算磁盘发生潜在扇区故障的可能性。如果磁盘有S个扇区，那么在任意时刻存在一个或多个潜在扇区错误的概率为$P_{LSE} = 1 - (1 - P_S)^S$。鉴于与不可恢复位错误相比，扇区错误是更常见的故障类型，因此在纳入硬错误的马尔可夫模型（见公式 3.22）中进行 MTTDL 计算时，可用P_{LSE}替代P_{UCER}。尽管这种变更能提高估算的准确性，但它没有考虑重建过程中的关键暴露区域，也没有考虑扇区错误的数量[30]。

7.2 模拟器

对于规范马尔可夫模型的所有扩展，先前讨论的马尔可夫模型仍存在的关键问题是故障率和修复率随时间变化。此外，有人认为这些模型无法很好地模拟多磁盘容错、不规则容错和扇区错误。鉴于这些原因，剩下最有效的方法就是仿真。标准仿真工具能够相当有效地处理时间依赖性，但要在合理的时间内得出结果并非易事（主要是因为稀有事件）。标准仿真工具使用两种方法来评估纠删码系统[63]的可靠性。一种基于不可靠度函数 $U(t)$ 的估计，另一种则基于平均数据丢失时间（MTTDL）。令 T_F 为表征故障时间的随机变量，T_i 为第 i 次迭代的停止时间，$\mathbf{1}_{T_F \le t}(t)$ 为示性函数，根据大数

定律可得

$$\hat{U}(t) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}_{T_F \le t}(t) \tag{7.3}$$

$$MTTDL = \frac{1}{N}\sum_{i=1}^{N} T_i \tag{7.4}$$

其中，假设仿真连续执行 N 次迭代。如前所述，在评估容错能力极高的系统时，前述仿真方法效率低下。这类仿真的主要缺点是，当失效概率极低时，获得准确结果所需的时间很长。通常，规避这一问题的主要方法是在标准模拟工具中，故障次数超过某个阈值后，提高稀有事件的发生频率。文献中通常将这种技术称为重要性抽样 [64]。其主要目的是在达到阈值后，提高观测到数据丢失事件的概率，同时降低（系统）方差。

已有许多更先进的基于仿真的可靠性估计器，因其高保真度的记录管理功能而闻名（跟踪发生故障的磁盘和扇区，并高效判定这些故障是否构成数据丢失事件）。其中之一是高保真可靠性（HFR）模拟器。有关更多信息及验证程序，请参阅参考文献 [30]。

参考文献

[1] A. Hoyland 和 M. Rausand，《系统可靠性理论》。纽约：Wiley，1994年。[2] B. Schroeder 和 G. A. Gibson，“现实世界中的磁盘故障：平均故障时间为 1 000 000 小时意味着什么？”载于 FAST 第五届 USENIX 会议论文集，2007年2月，第 1–16 页。[3] J. Yang 和 F.-B. Sun，“硬盘驱动器可靠性的全面综述”，载于年度可靠性与可维护性研讨会论文集，1999年，第 403–409 页。[4] E. Pinheiro、W. D. Weber 和 L. A. Barroso，“大型磁盘驱动器群体中的故障趋势”，载于 FAST 会议论文集，2007年，第 17–29 页。[5] J. Elerath 和 M. Pecht，“RAID 存储系统的增强型可靠性建模”，载于 DSN 国际会议论文集，英国爱丁堡，2007年6月，第 175–184 页。[6] H. E. Ascher，“一组数字并非数据集”，*IEEE Transactions on Reliability*，第 48 卷，第 2 期，第 135–140 页，1999年6月。[7] T. J. E. Schwarz、Q. Xin、E. L. Miller、D. D. E. Long、A. Hospodor 和 S. Ng，“大型归档存储系统中的磁盘擦洗”，载于 IEEE 计算机学会 MASCOTS 研讨会论文集，2004年，第 409–418 页。[8] S. S. Arslan 和 E. Zeydan，“ 大数据中心中重尾磁盘故障寿命的分布建模 ”，载于 *IEEE Transactions on Reliability*，第 70 卷，第 2 期，第 507–524 页，2021年6月。[9] Backblaze 的《硬盘数据与统计信息》。在线可用：www.backblaze.com/b2/hard-drive-test-data.html。[10] E. Zeydan 和 S. S. Arslan，“Cloud2HDD：面向云数据中心的云端大规模硬盘数据分析”，2020年第23届云、互联网与网络创新会议及研讨会（ICIN），法国巴黎，2020年。[11] Reed, I. S. 和 Solomon, G.，“特定有限域上的多项式码”，*SIAM*应用数学杂志，第 8 卷，1960年，第 300–304 页。[12] M. Blaum、J. Brady、J. Bruck 和 J. Menon，“EVENODD：一种可容忍 RAID 架构中双磁盘故障的高效方案”，*IEEE*计算机学报，44(2):192–202，1995年。[13] A. G. Tomislav、G. S. Kleiman、J. Leong、S. Sankar、P. Corbett 和 B. English，“用于双磁盘故障校正的行对角奇偶校验”，载于 *USENIX*文件与存储技术会议，2004年。[14] C. Huang 和 L. Xu，“STAR：一种用于校正三重存储节点故障的高效编码方案”，载于 *FAST'05*：第四届 *USENIX* 文件与存储技术会议论文集，美国加利福尼亚州伯克利，2005年。

[15] J. S. Plank 和 C. Huang，“教程：存储应用的纠删码”，在 *FAST–2013*：第 *11* 届 *Usenix* 文件与存储技术会议上展示的幻灯片，2013年2月。[16] R. G. Gallager，《低密度奇偶校验码》。马萨诸塞州剑桥：麻省理工学院出版社，1963年。[17] M. Luby，“LT码”，载于 ACM 计算机科学基础研讨会（FOCS）论文集，2002年。[18] S. S. Arslan，《Founsure 1.0：具备高效修复和更新功能的纠删码库》，*SoftwareX*，第 13 卷，第 1–12 页，2021年1月。[19] A. G. Dimakis、P. B. Godfrey、Y. Wu、M. J. Wainwright 和 K. Ramchandran，“分布式存储系统的网络编码”，*IEEE* 信息论汇刊，第 56 卷，第 9 期，第 4539–4551 页，2010年9月。[20] M. Pourmandi、A. E. Pusane、S. S. Arslan 和 E. Haytaoglu，“面向分布式存储系统的最小修复带宽LDPC码”，载于*IEEE*通信快报，2022年。DOI：10.1109/LCOMM.2022.3230263。[21] R. D. Cideciyan、S. Furrer 和 M. A. Lantz，“磁带数据存储的乘积码”，载于*IEEE*磁学汇刊，第 53 卷，第 2 期，第 1–10 页，2017年2月。[22] H. Shi、R. V. Arumugam、C. H. Foh 和 K. K. Khaing，“多层存储系统的最优磁盘存储分配”，载于*IEEE*磁学汇刊，第 49 卷，第 6 期，第 2603–2609 页，2013年6月。[23] O. Narmanlioglu、E. Zeydan 和 S. S. Arslan，“软件定义的下一代蜂窝网络中的服务感知多资源分配”，载于*IEEE Access*，第 6 卷，第 20348–20363 页，2018年。[24] S. S. Arslan，“利用拉丁方实现乘积码顺序介质记录的最优数据解码策略”，载于*IEEE*磁学汇刊，第 59 卷，第 1 期，第 1–11 页，2023年1月。[25] M. Rausand 和 A. Høyland，《系统可靠性理论》，John Wiley & Sons, Inc.，新泽西州霍博肯，第 2 版，2004年。[26] W. Burkhard 和 J. Menon，“磁盘阵列存储系统可靠性”，载于国际容错计算研讨会论文集，第 432–441 页，1993年。[27] W. A. Burkhard 和 P. D. Stojadinovic，“存储高效的可靠文件”，载于*1992*年冬季*USENIX*会议论文集，第 69–77 页，旧金山，1992年1月。[28] K. M. Greenan、J. S. Plank 和 J. J. Wylie，“平均无意义时间：MTTDL、马尔可夫模型与存储系统可靠性”，载于*USENIX*存储与文件系统热点问题研讨会论文集（HotStorage），第 1–5 页，2010年。[29] V. Venkatesan、I. Iliadis，“数据存储系统的一般可靠性模型”，载于第 *9* 届国际系统定量评估会议（*QEST 2012*）论文集，第 209–219 页，2012年。

[30] K. Greenan，《纠删码存储系统的可靠性与能效》，博士论文，加利福尼亚大学圣克鲁兹分校，2009年12月。[31] J. L. Hafner 和 KK Rao，“关于非 MDS 擦除码可靠性模型的说明”，技术报告 *RJ–10391*，IBM，2006年10月。[32] J.-F. Pâris、T. Schwarz, S.J.、A. Amer 和 D. D. E. Long，“适用于归档存储的高可靠性二维 RAID 阵列”，第 *31* 届国际计算机性能与通信会议论文集，第 324–331 页，2012年12月。[33] S. A. Weil、S. A. Brandt、E. L. Miller 和 C. Maltzahn。CRUSH：“副本数据的受控、可扩展、去中心化放置”，载于*2006*年 *ACM/IEEE* 超级计算会议（*SC’06*）论文集，佛罗里达州坦帕，2006年11月。[34] J. J. Wylie 和 R. Swaminathan，“高效确定基于异或的擦除码的容错能力”，DSN-2007：*IEEE* 国际可靠系统与网络会议，苏格兰爱丁堡，2007年6月。[35] J.-F. Pâris、A. Amer 和 T. J. E. Schwarz，“低冗余二维 RAID 阵列”，*2012*年国际计算、网络与通信会议数据存储技术与应用专题研讨会论文集，第 507–511 页，2012年1月至2月。[36]S. S. Arslan，“高效多维 MDS 奇偶校验保护分布式存储系统的冗余与老化”，*IEEE*设备与材料可靠性汇刊，第 14 卷，第 1 期，第 275–285 页，2014年3月。[37] J. J. Wylie 和 R. Swaminathan，“高效确定基于异或的擦除码的容错能力”，*DSN–2007*：*IEEE* 国际可靠系统与网络会议，苏格兰爱丁堡，2007年6月。[38] S. S. Arslan，“相关分布式 MDS 磁盘阵列单元的可靠性模型”，载于*IEEE Transactions on Reliability*，第 68 卷，第 1 期，第 133–148 页，2019年3月。[39] C. Huang、M. Chen 和 J. Li，“金字塔码：在可靠数据存储系统中以存储空间换取访问效率的灵活方案”，*ACM* 存储汇刊，9(1):1–28，2013年3月。[40] A. Wildani、T. J. E. Schwarz、E. L. Miller 和 D. D. Long，“防范归档系统中的罕见事件故障”，载于第 *17* 届 *IEEE/ACM* 计算机与电信系统建模、分析与仿真国际研讨会年会论文集（*MASCOTS’09*），第 1–11 页，英国伦敦，2009年9月。[41] Daniel Ford、Franis Labelle、Florentina I. Popovici、Murray Stokely、Van-Anh Truong、Luiz Barroso、Carrie Grimes 和 Sean Quinlna，“全球分布式存储系统的可用性”，载于第 *9* 届操作系统设计与实现研讨会论文集，（OSDI’10），加拿大温哥华，2010年10月。[42] S. Ramabhadran 和 J. Pasquale，“副本式分布式存储系统持久性的分析”，加州大学圣迭戈分校技术报告 *CS2007–0900*，2007年。

[43] S. S. Arslan、J. Peng 和 T. Goker，“一种用于载波辅助冷数据存储系统的数据辅助可靠性模型”，《可靠性工程与系统安全》第196卷，2020年4月，第106708号文章。[44] S. S. Arslan、J. Lee、J. Hodges、J. Peng、H. Le 和 T. Goker，“表头 CRC 校验失败和同步信号丢失情况下的 MDS 乘积码性能估算”，*IEEE Transactions on Device and Materials Reliability*第14卷，第3期，第921–930页，2014年。[45] A. Oprea 和 A. Juels，“重新审视磁盘擦洗”，载于 FAST，2010年，第57–70页。[46] J. Ryu 和 C. Park，“数据擦洗对存储系统可靠性的影响”，《电子信息通信学会信息与系统汇刊》第92卷，第9期，第1639–1649页，2009年。[47] S. Richards，“维护大型、高活跃度的磁带档案库”，载于第34届海量存储系统与技术国际会议。IEEE，2018年。[48] T. J. Schwarz、Q. Xin、E. L. Miller、D. D. Long、A. Hospodor 和 S. Ng，“大型归档存储系统中的磁盘擦洗”，载于 IEEE计算机学会第12届计算机与电信系统建模、分析与仿真年度国际研讨会，2004年（MASCOTS 2004）。会议录。IEEE，2004年，第409–418页。[49] M. Satyanarayanan，“文件大小与功能寿命研究”，载于《*ACM SIGOPS*操作系统评论》第15卷，第5期。ACM，1981年，第96–108页。[50] A. B. Downey，“文件大小分布的结构性成因”，载于 MASCOTS 2001：第九届计算机与电信系统建模、分析与仿真国际研讨会会议录。IEEE，2001年，第361–370页。[51] V. Ramaswami、K. Jain、R. Jana 和 V. Aggarwal，“用于网络性能评估的流量源重尾建模”，载于《计算智能、网络安全与计算模型》。施普林格，2014年，第23–44页。[52] W. Gong、Y. Liu、V. Misra 和 D. Towsley，“论 Web 文件大小分布的尾部”，载于*IEEE* 全阿勒顿通信、控制与计算年度会议会刊，第39卷，第1期。The University；1998，2001年，第192–201页[53] P. Constantopoulos、M. Doerr 和 M. Petraki，“长期数字保存的可靠性建模”，载于第九届*DELOS*卓越网络专题研讨会“数字仓储库：互操作性与通用服务”，希腊研究与技术基金会（FORTH），2005年。[54] Y. Han 和 C. P. Chan，“数字保存的系统可靠性建模：模型修改与四副本模型研究”，载于第五届数字对象保存国际会议（iPRES）会议录。大英图书馆，伦敦，2008年。[55] L. Ivanichkina 和 A. Neporada，“显性与潜在磁盘故障情况下分布式数据存储的可靠性模型”，《ARPN工程与应用科学期刊》，第10卷（20），2015年，C，第9150–9158页，2015年。

[56] J. L. Hafner 和 K. Rao，“关于非 MDS 擦除码可靠性模型的札记”，IBM研究报告 RJ–10391，2006年，2006年。[57] A. Dholakia、E. Eleftheriou、X.-Y. Hu、I. Iliadis、J. Menon 和 K. Rao，“一种适用于存在不可恢复错误情形的高可靠性 RAID 存储系统的新型盘内冗余方案”，*ACM* 存储汇刊（*TOS*），第4卷，第1期，第1页，2008年。[58] D. S. Papailiopoulos 和 A. G. Dimakis，“局部可修复码”，*IEEE* 信息论汇刊，第60卷，第10期，第5843–5855页，2014年。[59] A. G. Dimakis、K. Ramchandran、Y. Wu 和 C. Suh，“分布式存储网络编码综述”，*IEEE* 会刊，第99卷，第3期，第476–489页，2011年。[60] I. Iliadis、R. Haas、X.-Y. Hu 和 E. Eleftheriou，“高可靠性 RAID 存储系统中的磁盘擦洗与盘内冗余”，载于*ACM SIGMETRICS* 性能评估评论，第36卷，第1期。ACM，2008年，第241–252页。[61] L.N. Bairavasudaram，“部分磁盘故障的特征、影响与容忍度。博士论文”，威斯康星大学，美国，麦迪逊：ProQuest，UMI学位论文出版，2008年。[62] I. Iliadis、R. Haas、X.-Y. Hu 和 E. Eleftheriou，“高可靠性 RAID 存储系统中的磁盘擦洗与盘内冗余”，载于*SIGMETRICS’08*：*2008*年 *ACM SIGMETRICS* 计算机系统测量与建模国际会议会刊，第241–252页，美国纽约州纽约市，2008年。[63] V.F. Nicola、P. Shahabuddin 和 M.K. Nakayama，“高可信系统模型快速仿真技术”，*IEEE Transactionson Reliability*，50(3):246–264，2001年9月。[64] MarvinK. Nakayama 和 Perwez Shahabuddin，“用于估算大型高可靠性系统再生模型不可靠度的快速仿真方法”，《概率工程与信息科学》，18(3):339–368，2004年