# 大规模存储系统可靠性参数最优化分析

张林峰[1],谭湘键[2],杜 凯[3]
ZHANG Linfeng[1], TAN Xiangjian[2], DU Kai[3]

1.湖南农业大学 信息科学技术学院,长沙 410128
2.湖南农业大学 东方科技学院,长沙 410128
3.国防科技大学 计算机学院,长沙 410073
1.Information Science and Technology College of Hunan Agricultural University, Changsha 410128, China
2.Orient Science and Technology College of Hunan Agricultural University, Changsha 410128, China
3.School of Computer, National University of Defense Technology, Changsha 410073, China

**ZHANG Linfeng, TAN Xiangjian, DU Kai. Optimal reliability analysis for large scale storage systems. Computer Engineering and Applications, 2013, 49(1):112-119.**

**Abstract:** Data reliability are drawn much concern in large-scale storage systems built from thousands of storage devices, which highly depends on many inter-dependent system parameters, such as the replica placement policies, number of stored objects and so on. Previous work has discussed the impacts of these system parameters on reliability roughly and separately, and seldom provided their optimal values, nor mentioned their optimal combination. This paper presents a new object-based-repairing analytic model. Based on analyzing this model in three popular replica placement policies, it figures out the individual optimal values of these parameters at the beginning, and then works out their optimal combination. Compared with the existing models, this model is easier to solve while reaching more integrative and practical conclusions. These conclusions can directly and effectively instruct the designers to build more reliable storage systems.
**Key words:** data reliability; reliability model; large scale data center

**摘 要:** 在大规模存储系统中,数据的可靠性越来越受到人们的关注。已有的研究分析了在系统规模已知的条件下,某些系统参数,如副本分布策略、存储对象数目等,对可靠性的粗略影响,但较少提及它们的最优值或者最优组合。提出了一种基于对象粒度恢复的可靠性新模型;基于该模型,在分析三种主流的副本分布策略的基础上,分别计算出了各个系统参数的独立最优值及其组合最优值。与已有模型相比,该模型更易于求解,且获得了更加综合实用的最优值,这些最优参数值能直接有效地指导系统设计者构建更可靠的大规模存储系统。
**关键词:** 数据可靠性;可靠性模型;大规模数据中心
**文献标志码:** A **中图分类号:** TP393 doi:10.3778/j.issn.1002-8331.1106-0186

## 1 引言

当前许多依赖于大规模存储技术的应用已经在科学实验、电信通信、大规模互联网搜索引擎等诸多领域出现,如美国国家能源研究科学计算中心数据库包含了2.8 PB的信息[1];欧洲大强子对撞机实验产生数据速率高达1.5 GB/s的15 PB的数据[2]。这些系统往往是由存储节点集群构成,每个节点配备有CPU、内存和磁盘。其中具有代表性的系统是GFS、NASD、FAB[3]和Repstore[4]。大规模存储应用的另一个趋势是越来越多的应用需要存储WORM(Write Once Read Many,写一次读多次)类型数据[5]。因为磁盘带宽的增长速度远慢于磁盘空间的增长,由大量的存储组件构成的海量存储系统往往具有较高的故障率[6]和较长的恢复时间,因此为WORM类型数据构建高可靠的大规模存储系统是一个新的挑战,这是本文研究的主要问题。

当前已有许多研究关注数据可靠性问题[7-9],其中复制冗余机制是广泛使用的有效防止数据丢失的技术。如何在多个节点上放置副本,可以从两个方面显著地影响系统数据可靠性:多节点的脆弱性和恢复速度。基于不同的目

**基金项目:** 湖南省自然科学基金(No.07JJ5082)。
**作者简介:** 张林峰(1965—),男,副教授,主要研究方向为计算机网络与分布式计算研究;谭湘键,女,副教授,主要研究方向为计算机网络安全研究;杜凯,男,博士研究生,主要研究方向为分布式计算。
**收稿日期:** 2011-06-17 **修回日期:** 2011-08-25 **文章编号:** 1002-8331(2013)01-0112-08
CNKI出版日期:2011-10-24 http://www.cnki.net/kcms/detail/11.2127.TP.20111024.1015.090.html

的,已经提出并在实际系统中部署了多种副本分布策略,如GFS中的RANDOM,RAID中的PTN[7-8],并且,研究人员还定量地分析了这些策略对系统可靠性的影响。在三种策略中,假设 $n$ 个不同的对象存储在 $N$ 个存储节点上,且每个对象有 $K$ 个副本,每种副本放置策略用来指定副本和存储节点的映射关系。第一种策略是RANDOM,它随机的分布副本到节点上,GFS、FARSITE和RIO等系统都采用了该策略;第二种策略是PTN[10],它首先将所有对象副本分成多个组,然后将每一组放置到 $K$ 个连续的节点上,这种策略在RAID和Coda中采用;第三种策略是Q-rot,它将所有节点分成$K$个站点,每个站点是其他站点的副本。对每一种副本放置策略,一般都有两个基本的恢复操作:对象恢复(object repair)和对象重平衡(object rebalance)。前者是指在节点故障后将丢失的对象的副本临时在其他节点上生成;后者是指将临时生成的副本重新转移到新替换进来的节点上。

Lian和Chen[7-8]分析了系统容量、存储对象大小、磁盘和交换机带宽等系统参数如何影响系统可靠性,但是仅粗略地讨论了影响的趋势而没有提供这些参数的最优值,而且没有分析它们的联合影响和组合最优值。另外,为了获得精确的可靠性,某些模型过于复杂以至于难以计算出每个系统参数的最优值。

为了克服以上种局限,针对系统设计者依据应用需求确定系统规模后,面临难以确定其他相关系统参数来构建高可靠系统的问题,本文旨在计算出优化的系统参数及其组合以设计出高可靠的系统。

## 2　数据可靠性定义

如何放置对象副本到众多的存储节点上会显著地影响系统可靠性,因此许多研究都关注这个问题。

数据可靠性一般指在丢失第一个存储对象前一个存储系统提供服务的时间。本文,用平均数据丢失时间(Mean Time To Data Loss,MTTDL)来表示数据可靠性。在存储系统中,单个对象的可靠性不同于系统可靠性,因为每个节点一般存储多个对象副本。所以,将通过计算在三种不同放置策略中单个对象的可靠性和多个对象位置的相关性来刻画系统的可靠性。

### 2.1　存储对象可靠性$MTTDL_O$定义

存储对象可靠性$MTTDL_O$定义为单个对象的所有副本全部丢失的平均时间。$MTTDL_O$受对象故障率和对象恢复速度两个因素的影响,前者由磁盘故障率和对象放置策略确定,后者受对象平均大小和并行恢复带宽影响(参与恢复的节点和主干恢复网络带宽)。

### 2.2　系统可靠性$MTTDL_S$定义

假设系统中独立对象的数目是 $n_i$,定义系统可靠性$MTTDL_S$为:

$$MTTDL_S = \frac{MTTDL_O}{n_i}$$

独立对象定义为:如果两个对象的每一个副本都在同一个节点上,或者说这两个对象具有相同的分布,那么这两个对象是相关的,否则它们是相互独立的,因为其中一个的故障不会影响另外一个。使用这种较弱的独立性定义得到的是较为严格的$MTTDL_S$定义。基于上述定义,可以得到如表1的三种策略的独立对象数目公式,见表1。

**表1　三种策略的独立对象数目公式**

| | RANDOM | Q-rot | PTN |
|---|---|---|---|
| $n_i$ | $\min(m\times\lceil N/K\rceil, C_N^K)$ | $\min(m\times\lceil N/K\rceil, \lceil N/K\rceil^2)$ | $N/K$ |

表1中,$C_N^K$ 表示从$N$个节点中选择$K$个来放置$K$个副本的组合数,其定义参见表2。

下面基于马尔可夫模型计算$MTTDL_O$的基础上,依据上面的公式进一步得到$MTTDL_S$。

## 3　系统可靠性分析

从第2章的分析中可以看出,系统可靠性主要与三个因素有关:存储对象的故障概率、修复速度和两者的相关性。本章在定义一些系统参数的基础上,将首先独立分析前面两个因素,然后创建Markov模型来描述对象故障和修复过程。

### 3.1　系统参数定义

表2列出了下文将用到的一些系统参数。其中前四个参数不但会显著影响系统故障率和修复速度,而且计算$MTTDL_S$时其最优值是相关的。因此,将重点关注它们对的不同影响。表2中余下的参数都视为常数,因为它们的影响是简单而且直接的。比如系统数据容量$S$越小,或者副本数$K$越大,或者节点故障率$\lambda$越低,系统可靠性就越高。所以在下文的分析中,$K$、$\lambda$和$S$的值将设置为表2中的默认值。

**表2　系统参数表**

| 参数 | 定义 | 默认值 |
|---|---|---|
| $N$ | 总节点数 | 变量 |
| $m$ | 单个节点上存储的评价对象个数 | 变量 |
| $B$ | 网络带宽 | 变量 |
| $b$ | 单个节点IO带宽 | 变量 |
| $q$ | $B$和$b$用于对象恢复的比例;<br>$(1-q)$是用于对象重平衡的比例 | 90% |
| $K$ | 每个对象的副本数 | 3 |
| $\lambda$ | 节点故障率 | 1/3(1/year) |
| $S$ | 数据总量 | 1 PB |
| $n$ | 不同对象总数 | $N\times m/K$ |
| $s$ | 单个对象平均大小 | $S/(N\times m)$ |
| $n_i$ | 独立对象数目 | 参见表1 |
| $n_f$ | 一天内平均故障节点数目 | $\lceil N\lambda/365\rceil$ |
| $n_r$ | 需要并行恢复的对象数目 | $m\times n_f$ |

### 3.2　脆弱性分析:故障率

在某些应用中,一个用户请求只访问一个对象,比如在YouTube上看某个短片;在另外一些应用中,一次请求需要访问多个对象,比如数据库应用和编译具有多个源代码

文件的工程。所以从单个请求访问对象的数目的角度,需要分析单个和多个对象的故障率(Failure Probability,FP),及其对系统脆弱性的影响。

因为一般任意对象的多个副本不会存储到同一个节点上,那么一个对象丢失当且仅当存储该对象的 $K$ 个副本的 $K$ 个节点都出现故障了。一个对象的故障率就是所有 $K$ 个节点都出现故障的概率,所以可以得到在PTN和Q-rot中单个对象的故障率是 $\frac{p^3K!}{K(N-1)(N-2)\cdots(N-K+1)}$,在RANDOM中是 $p^3$(其中,$p$ 是单个节点的故障率)。一般而言,$N$ 远大于 $K$,所以PTN和Q-rot中的单个对象的故障率小于RANDOM中。该结果表明,就单个对象而言,规则的放置对象往往比随机的更鲁棒。

多个对象的故障率不能直接计算出来,因为每个请求访问的对象数目不是固定的,所以通过仿真的方式来获得。通过变化系统中节点数 $N$,总的对象数目 $n$ 和对象访问比例 $p$ 来进行仿真,以确定影响趋势。仿真的第一个结果是当 $p$=0.1时,PTN具有最低的固定值是0.2,而RANDOM和Q-rot更差。所以从可用性的角度看,在三种分布中PTN是最高的。另一组仿真结果:(1)$n$ 越小,RANDOM和Q-rot可用性越高;(2)$N$ 越小,PTN可用性越高;(3)$N$ 和 $n$ 越小,RANDOM和Q-rot可用性越高。

从上面的分析可以看出,$N$ 和 $n$ 影响了单个和多个对象的故障率。另外,如果不考虑恢复,RANDOM和Q-rot是非常脆弱的,因此,恢复速度必须重点考虑。

### 3.3　可恢复性分析:恢复速度

从2.1节中可知,虽然最小的故障单位是存储节点,但在恢复过程中,恢复的最小单位是存储对象。在恢复过程中,一般会以相同的优先级同时恢复多个对象,其分配的恢复带宽也是相同的。因此,一个对象的恢复时间等于其大小除以平均恢复带宽(因为针对WORM类型数据,维护数据一致性的开销是0)。平均修复带宽(Repairing Bandwidth,RPB)由四个因素确定:参与恢复的源节点的总带宽、目标节点的总带宽、交换机的带宽和并发恢复对象的数目。其计算公式如下:*rpb*=min(源节点总带宽,目标节点总带宽,网络总线带宽)/(需要恢复对象副本数目),因为这三者中的最小值才是真实的带宽。

并发恢复的对象数目不是一个常数,但是可以通过如下的方式来估计其平均值:每天平均故障节点数 $n_f$ 乘以每个节点上的对象数 $m$,即 $n_r=m\times n_f=m\times\lceil N\lambda/365\rceil$。

前述的三种副本分布方式会导致不同的恢复带宽,可以分别如下计算。在PTN中,最大的恢复的源节点带宽是 $(K-1)bq$,目标节点带宽是 $(N-n_f)bq$。而一般而言,由于 $B>>K-1$,$(N-nf)>>K-1$,因此在中对象恢复带宽是:

$$rpb_{\text{PTN}}=\frac{\min((K-1)bq,(N-n_f)bq,Bq)}{n_r}=\frac{(K-1)bq}{n_r}\quad(1)$$

在RANDOM和Q-rot中,最优的恢复带宽是:

$$rpb_{\text{Q-rot}}=rpb_{\text{RANDOM}}=\min(\frac{bq(N-n_f)}{n_r},\frac{Bq}{n_r},bq)\quad(2)$$

如文献[9]中所述,在恢复过程中,当故障节点数较多时,Q-rot使用了几乎所有完好的节点作为源节点。根据这个结论,用 $(N-n_f)$ 来表示源节点的数目。另外,在式(2)中,$b$ 和 $B$ 都是双工带宽。

另一个需要考虑的因素是对象重平衡速度。如同对象恢复速度一样,只有重平衡带宽需要考虑。三种副本放置策略的重平衡带宽具有如下相同的计算公式:

$$rbb_{\text{PTN}}=rbb_{\text{Q-rot}}=rbb_{\text{RANDOM}}=\min(b/m,b(1-q))\quad(3)$$

### 3.4　Markov可靠性模型

根据3.2节和3.3节中的分析结果可知,$N$ 和 $n$ 对系统的脆弱性和可恢复性两者影响是对立矛盾的,而数据可靠性是脆弱性和可恢复性两者的结合。因此,基于一个新的可靠性模型来分析它们的综合影响是必须的。

在2.2节中,已经定义可以计算对象可靠性来导出的系统可靠性。因此,基于Markov模型在可靠性分析方面的有效性[11],在图1中构建了一个连续时间的Markov模型。这种基于对象的模型描述了对象副本的故障和恢复过程。

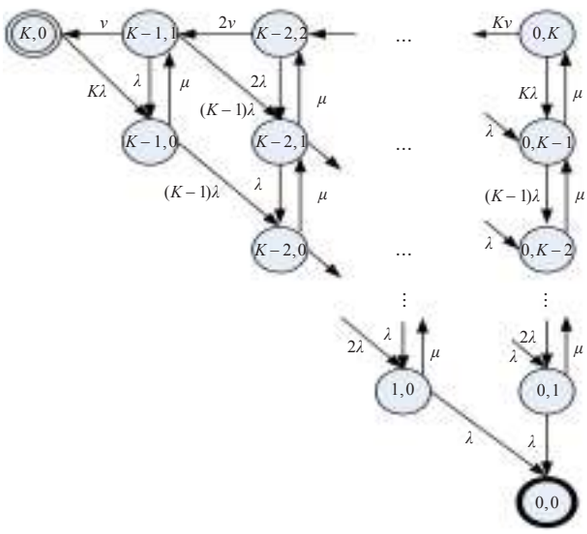


**图1　基于对象的数据恢复过程**

该过程中状态由 $(k,i)$ 定义,其中 $k$ 表示某对象保留在原始节点上的副本的数目,$i$ 表示临时恢复存储到某些节点上等待重平衡到新替换上线的新节点上的该对象的副本的数目。初始状态表示 $K$(在表2中定义的)个副本都在初始节点上。唯一的吸收状态(0,0)表示该被考察对象的所有副本都丢失了。$MTTDL_o$ 即为从初始状态 $(K,0)$ 到吸收状态(0,0)的平均时间。

为了量化计算 $MTTDL_o$,所有状态之间的转换速度是必须确定的。图1中 $\lambda$ 是副本的出现故障的速度,和表2中的节点故障速度一样。$\mu$ 是副本恢复速度,可以通过3.3节和表2中定义的两个参数计算得出,即 *rpb*/s。类似的 $v$ 表示副本重平衡速度,即 *rbb*/s。

在所有转换速度都确定后,$MTTDL_o$ 可以按照参考文献[8]中的方法求解状态转换矩阵得到。由于 $MTTDL_o$ 的表达式过于繁杂,在这里就不展开其表达式。

与文献[8]中的模型相比，本文的模型更简单，原因是忽略了替换故障节点的细节。之所以如此考虑，是因为对象副本的恢复是在故障节点替换之前完成的，所以故障节点替换的速度不会直接影响恢复速度，而仅仅影响重平衡速度。文献[8]中的模型关注节点的替换和对象的恢复，而模型着重研究在假设平均故障节点数时对象的恢复情况。选择这样的观察角度的好处是模型的状态空间大为减少，本文状态空间大小是 $O(K^2)$，而文献[8]是 $O(NK)$。因为 $N>>K$，而且在一个大规模的存储系统中 $N$ 往往大于1 000，此时求解这样一个庞大的状态转换矩阵是十分复杂的。在本文模型中，由于忽略了一些信息，如当前故障节点数目，导致可靠性的结果没有那么精确。但是应用场景中，更加关注系统参数的最优值而不是精确的系统可靠性本身。下面将评价和分析系统参数的影响。

## 4　最优系统可靠性参数

基于前文中的副本恢复和重平衡公式，以及马尔可夫模型，本章将通过分析在不同条件下各个参数对MTTLS的独立的影响，计算出四个参数 $m$、$B$、$b$ 和 $N$（同3.1节介绍的，其他参数的影响将不予分析）的最优值。通过分析 $MTTL_S$ 的表达式发现，对象恢复速度 $rpr$ 和独立对象数目 $n_i$ 是其两个关键性因素。$n_i$ 可以从表1中得到，三种放置策略的 $rpr$ 可以计算如下（近似条件是 $N-n_f\approx N$ 和 $\lceil N\lambda/365\rceil\approx N\lambda/365$）：

$$rpr_{\text{Q-rot}}=rpr_{\text{RANDOM}}=\frac{\min(\frac{\min(bq(N-n_f),Bq)}{n_r},bq)}{\frac{S}{Nm}}\approx$$

$$\frac{\min(\frac{\min(bq(N-n_f),Bq)}{mN\lambda/365},bq)}{\frac{S}{Nm}}\approx$$

$$\frac{365}{S}\min(\frac{Nbq}{\lambda},\frac{Bq}{\lambda},\frac{Nmbq}{365}) \tag{4}$$

$$rpr_{\text{PTN}}=\frac{mn_f}{\frac{S}{Nm}}\approx\frac{365(K-1)bq}{S\lambda} \tag{5}$$

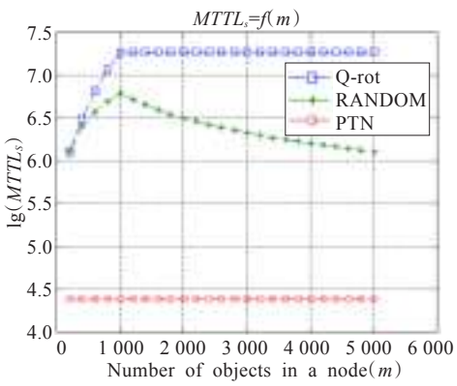


**图2**　$MTTL_S$ 关于单个节点上对象数目 $m$ 的曲线（$b$=20 MB/s，$B$=30 GB/s，$N$=1 023，$n_f$=1（$Bq>Nbq$ and $365/\lambda>N/K$））

公式(4)显示了参数 $m$、$B$、$b$、$N$ 如何影响RANDOM和Q-rot中的恢复速度；公式(5)说明了 $b$ 对 $rpr_{\text{PTN}}$ 的影响。下面将依据前文的公式和马尔可夫模型的求解结果计算出 $m$、$B$、$b$、$N$ 的独立最优值。尽管 $rpr$ 的计算是一个近似值，但是 $N$ 和 $m$ 的最优近似值数量级仍然具有现实指导意义。

### 4.1　$m$ 的最优值

根据公式(4)，为了得到存储在单个节点上的对象数目的最优值，需要分两种情况讨论：$Nbq/\lambda<Bq/\lambda$ 和 $Nbq/\lambda\geq Bq/\lambda$。前者在图2和3中显示，后者在图8中显示。

在图2中，因为 $Nbq/\lambda<Bq/\lambda$，所以 $Nmbq/365$ 和 $Nbq/\lambda$ 的大小关系决定了曲线走向。当 $Nmbq/365<Nbq/\lambda$ 时，$rpr_{\text{Q-rot}}$ 和 $rpr_{\text{RANDOM}}$ 随 $m$ 的增大而增大，$MTTL_S$ 同样。当 $Nmbq/365\geq Nbq/\lambda$ 时，$rpr_{\text{Q-rot}}$ 和 $rpr_{\text{RANDOM}}$ 达到最大值后保持不变，而 $MTTL_S$ 达到最大值后不变或者变小。从图中可以看出，随 $m$ 的增大，独立对象的数目 $n_i$ 增大，$MTTL_S$ 一直减小直到 $n_i$ 达到最大值时才保持不变。由此，可以得到的如下最优值（当 $Nmbq/365=Nbq/\lambda$ 时）：

$$m_{\text{optimal}}=\frac{Nbq/\lambda}{Nbq/365}=\frac{365}{\lambda} \tag{6}$$

在图3中，因为 $Bq<Nbq$，那么 $Nmbq/365$ 和 $Bq/\lambda$ 的大小关系就决定了 $rpr_{\text{Q-rot}}$ 和 $rpr_{\text{RANDOM}}$。除了 $m$ 的最优值不一样外，$MTTL_S$ 随 $m$ 而变化的方式和图6和图7一样：

$$m_{\text{optimal}}=\frac{Bq/\lambda}{Nbq/365}=\frac{365B}{Nb\lambda}=\frac{365}{\lambda}\times\frac{B}{Nb} \tag{7}$$

由图2和图3的结论可以总结成如下的表达式：

$$m_{\text{optimal}}=\frac{365}{\lambda}\times\min(\frac{B}{Nb},1) \tag{8}$$

### 4.2　$N$ 的最优值

和4.1节的讨论类似，为了得到总节点数 $N$ 的最优值，需要依据 $Nmbq/365$ 和 $Nbq/\lambda$ 的大小关系，分成以下两种情况进行分析：$Nmbq/365\leq Nbq/\lambda$（参见图4）和 $Nmbq/365>Nbq/\lambda$（参见图5）。在图4中有 $m\leq 365/\lambda$，当 $Nmbq/365<Bq/\lambda$ 时，随 $N$ 的增大，$rpr_{\text{Q-rot}}$，$rpr_{\text{RANDOM}}$ 和 $MTTL_S$ 单调增大；当 $Nmbq/365=Bq/\lambda$ 时，它们都达到最大值。因此，$N$ 的近似最

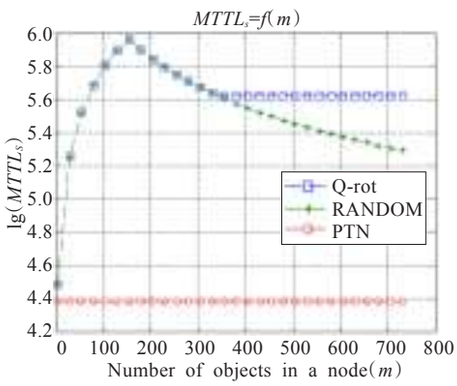


**图3**　$MTTL_S$ 关于单个节点上对象数目 $m$ 的曲线（$b$=20 MB/s，$B$=3 GB/s，$N$=1 023，$n_f$=1（$Bq<Nbq$））

优值是：

$$N_{\text{optimal}}=\frac{Bq/\lambda}{mbq/365}=\frac{365B}{Nb\lambda}=\frac{365}{m\lambda}\times\frac{B}{b} \tag{9}$$

在图5中，$MTTL_S$的曲线和图4基本相似。当$Nbq/\lambda=Bq/\lambda$时，$MTTL_S$达到最大值，此时得到$N$的最优值$N_{optimal}$是：

$$N_{\text{optimal}}=\frac{Bq/\lambda}{bq/\lambda}=\frac{B}{b} \tag{10}$$

综合式(9)和(10)，可以得到$N$的最优值是：

$$N_{\text{optimal}}=\frac{B}{b}\times\max(\frac{365}{m\lambda},1) \tag{11}$$

上式表明$N_{\text{optimal}}$仅由$B/b$和$m\lambda$确定。

### 4.3　$B/b$的最优值

根据$rpr_{\text{Q-rot}}$和$rpr_{\text{RANDOM}}$的公式可知，当$B$和$b$视作可调变量时，因为$B/b$是一个整体出现的因子，所以选择$B/b$作为一个变量来分析$MTTL_S$比较合适。类似于4.1节和4.2节的分析，通过比较$Nmbq/365$和$Nbq/\lambda$的大小，将讨论两种情况：$Nmbq/365\leq Nbq/\lambda$(图6)和$Nmbq/365>Nbq/\lambda$(图7)。

如图6所示，当$Nmbq/365<Bq/\lambda$时，$rpr_{\text{Q-rot}}$、$rpr_{\text{RANDOM}}$和$MTTL_S$随$N$的增大单调增大；当$Nmbq/365\geq Bq/\lambda$时，它们都达到最大值且保持不变。所以$B/b$的最优值如下：

$$B/b_{\text{optimal}}\geq\frac{Nm\lambda}{365} \tag{12}$$

在图7中，$MTTL_S$的曲线和图6类似。当$Nbq/\lambda=Bq/\lambda$时，$MTTL_S$达到最大值：

$$B/b_{\text{optimal}}\geq N \tag{13}$$

综合式(12)和(13)，可以得到：

$$B/b_{\text{optimal}}\geq N\times\min(\frac{m\lambda}{365},1) \tag{14}$$

## 5　$N$、$m$、$B/b$最优系统可靠性参数组合

第4章在基于其他参数是常量的基础上，得到了各个参数的独立最优值。本章将首先分析这些参数如何联合影响系统可靠性，基于此找到理论上的联合最优值，然后讨论如何在真实系统中使用这些最优值。

### 5.1　联合最优值

从第4章的结论可以看出，Q-rot往往具有最好的可靠性，因此，出于篇幅考虑，本章仅分析Q-rot中的参数的联合最优值。

假设是$MTTL_S=f(B/b)$是$MTTL_S$关于$B/b$的函数，从4.3节可以发现$MTTL_S=f(B/b)$是一个单调非减函数。因此，为了计算联合最优值，先将$B/b$固定为一个常数，然后分析$m$和$N$对$MTTL_S$的联合影响，最后得到最优组合值。

图8中，曲线$(m,MTTL_S)$在$N$变化时表现了与4.1节相同的变化趋势，这表明$N$不干扰$m$对$MTTL_S$的影响。与此类似，在图9中，$m$也不改变$N$对$MTTL_S$的影响。

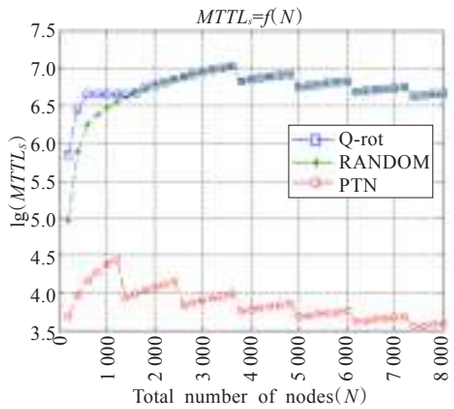


**图4**　$MTTL_S$关于节点总数$N$的曲线

($b$=20 MB/s，$B$=30 GB/s，$m$=500($Nmbq/365\leq Nbq/\lambda$))

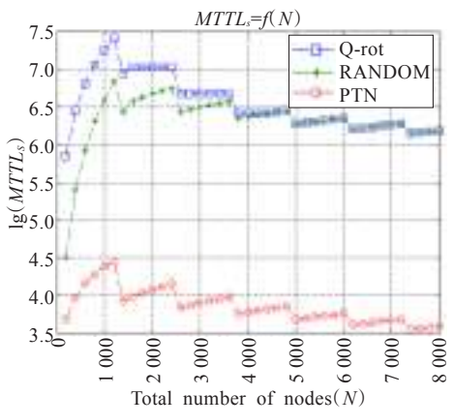


**图5**　$MTTL_S$关于节点总数$N$的曲线

($b$=20 MB/s，$B$=30 GB/s，$m$=1 500($Nmbq/365>Nbq/\lambda$))

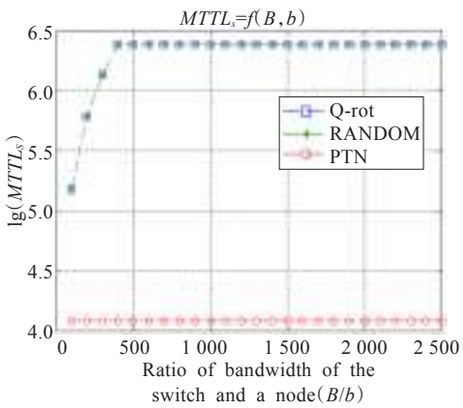


**图6**　$MTTL_S$关于$B/b$的曲线

($N$=2 048，$m$=200，$b$=20 MB/s($Nmbq/365\leq Nbq/\lambda$))

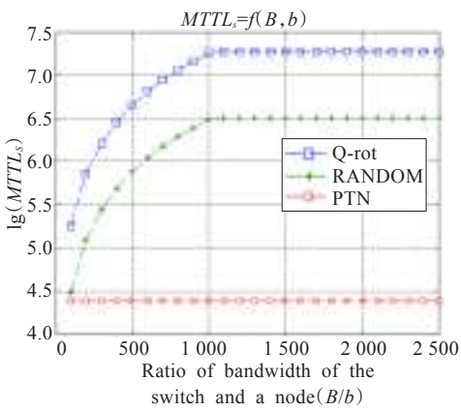


**图7**　$MTTL_S$关于$B/b$的曲线

($N$=1 024，$m$=2 000，$b$=20 MB/s($Nmbq/365>Nbq/\lambda$))

从这些结论可以推导出$m$和$N$对$MTTL_S$的影响是正交的。类似地，可以发现$B/b$也不干扰$m$和$N$对$MTTL_S$的影响，反之亦然。

根据4.1节的公式有$m_{optimal}\leq 365/\lambda$，结合4.2节公式得到$N_{optimal}=365B/mb\lambda$。类似地，由4.2节$N_{optimal}\geq B/b$，结合4.1节结论可以得到$m_{optimal}=365B/Nb\lambda$。这些结论在图10中得到验证，该图由图8和9合成而成（因为第4章开头取近似值，所以其中的数值不是完全精确的）。由此得到的另一个结论是$B/b_{optimal}\geq Nm\lambda/365$。

据此，关于组合最优值，可以得到如下结论：

$$N_{optimal}=365B/mb\lambda \tag{15}$$

$$m_{optimal}=365B/Nb\lambda \tag{16}$$

$$B/b_{optimal}\geq Nm\lambda/365 \tag{17}$$

由于$MTTL_S$函数表达式的复杂性，很难通过解析的方式确定参数的最优值，所以通过多次计算$MTTL_S$的方式来确定，结果发现$N$的最优值$N_{optimal}$总是接近$\lceil 365/\lambda\rceil$，$N$的其他的次优值都是$\lceil 365/\lambda\rceil$的倍数。这个结论表明$N_{optimal}$仅仅取决于节点的故障率，据此可得$m_{optimal}=B/b$。$N$、$m$、$B/b$的联合最优值能够直接指导设计者构建高可靠的存储系统。

### 5.2　真实系统中最优值应用

在5.1节已经得到了$N$、$m$、$B/b$的组合最优值。在实际系统中，还有一些其他的因素会影响设计，这些因素给这些系统参数带来了更多的限制。

为了让这些理论联合最优值更好地符合这些限制的要求，对其最优值进行了扩展：

（1）从IO性能角度考虑，$N$越大意味着IO性能越高。据此，$N_{optimal}$可以是$\lceil 365/\lambda\rceil$的倍数，而不必局限于仅仅等于$\lceil 365/\lambda\rceil$，这样可以得到一些次优的$MTTL_S$。

（2）$m_{optimal}$可以大于$B/b$，$m_{optimal}=B/b$（在图9中是1 210）是一个较小的值，在实际系统中会被轻易超出。一个替代的方案是将一组对象打包成一个组，每组看做一个大的对象按照分布策略进行分布存储。因此可以通过保持组的数目接近$B/b$来达到最优的$MTTL_S$。此时，不论多少对象都可以存储在系统中（组中的对象可以持续追加）。

## 6　实验测试

实验将通过和已有的模型关于参数最优值的结论进行比较，来评价本文的工作。Lian[7]和Chen[8]研究的主要问题和本文接近，因此与已有研究的异同点将在此进行深入探讨。

文献[7-8]都基于RANDOM分布策略进行了分析。本文模型在图11也分析了RANDOM策略，其结论显示，当其他参数和文献[7-8]相同的时候$m$和$N$如何共同影响$MTTL_S$，且得出了同第5章类似的结论：$N_{optimal}$大约是1 200，接近$\lceil 365/\lambda\rceil$，$m_{optimal}$接近$B/b$。

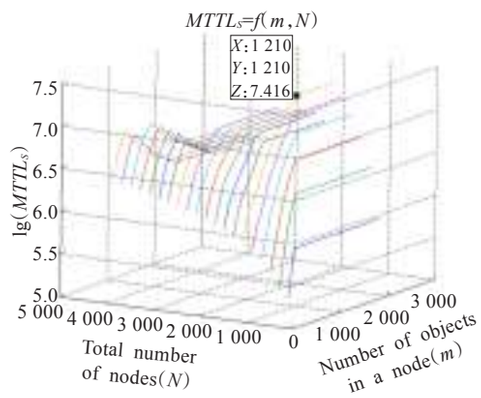


**图8　$MTTL_S$关于$m$和$N$的曲线（从$m$的视角）**
（$b$=20 MB/s，$B$=30 GB/s）

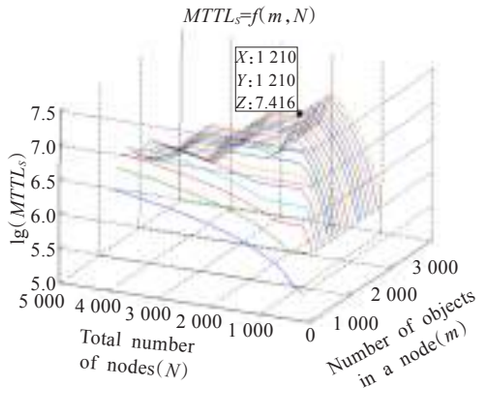


**图9　$MTTL_S$关于$m$和$N$的曲线（从$N$的视角）**
（$b$=20 MB/s，$B$=30 GB/s）

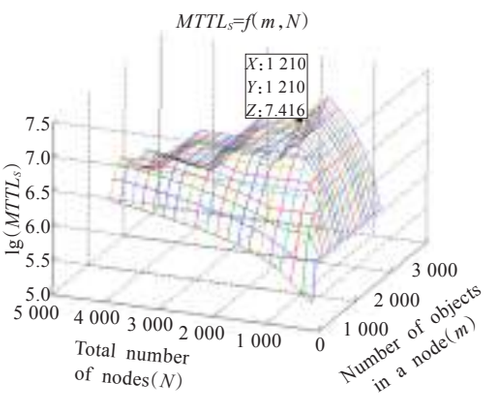


**图10　$MTTL_S$关于$m$和$N$的曲线（从系统的视角）**
（$b$=20 MB/s，$B$=30 GB/s）

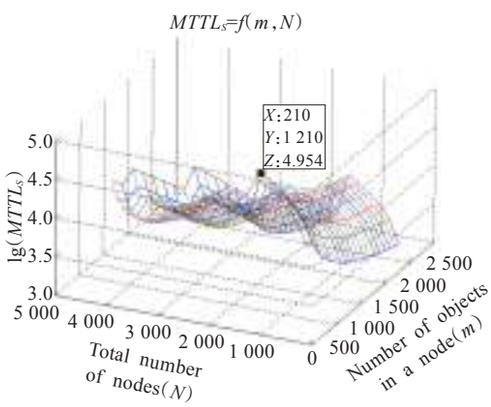


**图11　$MTTL_S$关于$m$和$N$的曲线（从系统的视角）**
（$b$=20 MB/s，$B$=3 GB/s，$S$=3 PB）

在文献[7-8]中,仅仅对单个节点上的对象数目 $m$,或者对象的平均大小 $s$ 进行了量化分析,因此只能将图11中的 $m$ 值与文献[7-8]进行比较,比较的结果见图12。这里 $N$=1 024是其模型中的公共值,该 $N$ 值接近本文的 $N_{optimal}$,同时导致 $m$ 也接近本文的 $m_{optimal}$。

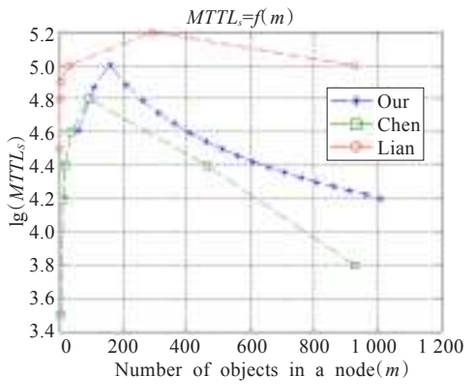


**图12　本文模型与Chen和Lian的比较**

($b$=20 MB/s,$B$=3 GB/s,$S$=3 PB,$N$=1 024)

图12是三个模型关于 $m$ 的比较结果。从图12可以得到如下结论:

在三个模型中,$m_{optimal}$ 分别是93、160、293,这些值都接近。这表明关于 $m_{optimal}$ 的结论几乎和文献[7-8]的一样;另外,这些曲线的形状基本相同。

尽管上面的分析表明,本文模型可以得到与已有的模型类似的结论,但是由以下三点可以说明本文的研究是独特的和必要的。

(1)为什么更简单的模型可以得到比已有复杂模型类似的结论。本文基于对象恢复的模型和文献[8]的差别在于故障节点的数目。这里采用了平均故障节点数目,而Chen基于节点恢复的视角利用了当前故障节点的数目。实际上,故障节点的平均数目是日常故障节点的统计结果,这足够计算另一个统计平均值 $MTTL_S$。这是为什么本文结论和文献[8]接近的主要原因。

(2)为什么文献[7-8]没有量化节点总数 $N$ 对系统可靠性的影响。在文献[7]中,因为在其简单的基于节点恢复的模型中没有包含公式(2)中的第三项和故障节点数这些参数,所以 $N_{optimal}$ 不能计算得到。在文献[8]中,尽管其模型可以计算出更为精确的 $MTTL_S$,但是其复杂的恢复带宽公式和动态的恢复节点数使得其难以计算出 $N_{optimal}$;另外,文献[8]中的状态空间的规模是 $O(NK)$,所以很难通过直接求解模型计算出 $N_{optimal}$,因为当 $N$ 非常大的时候,由于状态空间是随着 $N$ 呈线性增长的,这对很大规模状态空间进行求解是很困难的事情。

(3)关于联合最优值的问题,这一点在文献[7-8]中都没有讨论。通过分析观察 $MTTL_S$ 关于 $N$ 和 $m$ 的曲线,得到了最优组合值。因其提供了一个完整的设计所需要考虑的组合参数而非单个最优值,所以本文结论在设计真实系统时更实用。

## 7　相关工作

在大规模存储系统中数据可靠性是不容忽视的问题。较早的研究基于独立的指数分布的磁盘故障率,使用马尔可夫模型分析了RAID的 $MTTD_L$。当存储规模达到更大时,系统故障率增高,恢复速度成为关键因素,如何在拓展RAID思想的基础上,从这两个方面来提高数据可靠性,成为研究人员关注的重点。本文的研究就属于这类。

Xin等[12]研究了如何在RAID内部和RAID之间实现分布式恢复,但较少关注分布策略对于恢复速度的影响。Ramabhadran等分析了在长期运行的副本系统中单个对象的可靠性(类似于2.2节的MTTDLO),但没有考虑可用恢复带宽的影响。

Øystein Torbjørnsen提出了分布策略,并且基于节点恢复的马尔可夫模型分析了诸多影响系统可用性的因素。但他们主要关注数据库的读写访问模式如何影响系统的可用性(availability),这一点与本文关注可靠性(reliability)不同。

Lian的研究[7]与本文的接近。他们重点研究了两种不同的分布策略对可靠性的影响,并基于简单的模型量化分析了某些参数。Chen[8]的研究是文献[7]的扩展。基于一个复杂的节点-对象恢复模型,分析了其他一些因素的影响,如拓扑敏感的副本放置策略、主动副本机制等。与这些研究不同的是,本文不但研究了系统参数如何影响可靠性,而且计算出了这些参数面向可靠性的最优值。另外,本文分析了这些参数的组合最优值,而不仅仅是独立最优值,这也没有在上述文献中讨论。

大量研究是关于大规模系统的可用性,而非可靠性。Yu[13]、穆飞[14]等研究了在广域网中不同副本分布策略下的多对象操作可用性,这点与3.2节相关;Nath[15]等分析了广域网中针对联合故障的基于纠删码技术的数据可用性;Douceur利用动态副本分布策略提高了文件的整体可用性;Renesse[16]研究了DHT和随机分布策略对分布式存储系统的可用性的影响。但他们大多采用仿真的方法,而且没有考虑可用网络带宽的影响。

## 8　结论与下一步的工作

设计可靠的大规模存储系统面临许多挑战。其中之一是在系统规模确定的情况下,计算出一些基本的系统参数,如存储对象的数目、存储节点数目、网络带宽、节点IO带宽等。与以前的工作相比,本文的研究在两方面有所突破。首先,提出了一个新的基于对象的马尔可夫模型,基于该模型量化分析在系统规模已知的情况下,三个常用的副本放置策略中系统参数对可靠性的影响。这些系统参数包括存储节点总数、对象总数、交换机带宽、磁盘带宽等。相对于已有的较为复杂的模型,这种基于对象的简洁模型一方面因其较小规模的状态转换矩阵而易于求解,另一方面便于获得更加综合和实用的结果。其次,提出了一个两阶段的分析过程。第一阶段是在固定其他参数的前提下,通过独立分析各个参数的影响,找出相对精确的最

优值;第二阶段是在所有参数都可变的情况下,通过分析它们的综合复杂的影响,来得到这些参数的最优组合值。分析结果表明,这些最优值确实存在且具有简单的形式,这样就可以极大地方便高可靠系统的设计。一般而言,越高的数据可靠性意味着越高的成本,而且,系统IO性能需求往往和数据可靠性发生冲突。因此,今后的工作将研究如何在大规模系统中平衡更多的因素:成本、IO带宽、可靠性等,为系统设计者提供最优参数值。

## 参考文献:


[1] Business Intelligence Lowdown Company.The top 10 largest databases[EB/OL].(2007-02-18)[2011-04-01].http://www.businessintelligencelowdown.com/2007/02/top_10_largest_.html.

[2] Abadie L,Badino P,Baud J,et al.Grid-enabled standards-based data management[C]//Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies(MSST 2007),2007.

[3] Saito Y.FAB:building distributed enterprise disk arrays from commodity components[C]//Proceedings of the 11th ASPLOS,Oct 2004.

[4] Zhang Z.RepStore:a self-managing and self-tuning storage backend with SmartBricks[C]//Proceedings of the 1st IEEE International Conference on Autonomic Computing,May 2004.

[5] Reference information:the next wave[R].The Enterprise Storage Group,2002-06.

[6] Chen Wei.Trends and challenges in large-scale data center availability[C]//Proceedings of the Panel on SRDS 2007,2007.

[7] Lian Q,Chen W,Zhang Z.On the impact of replica placement to the reliability of distributed brick storage systems[C]//Proceedings of the 25th ICDCS,Jun 2005.

[8] Chen Ming,Chen Wei,Liu Likun,et al.An analytical framework and its applications for studying brick storage reliability[C]//Proceedings of the IEEE SRDS 2007,2007.

[9] 刘仲,李宗伯.基于对象分组的集群存储系统可靠性分析模型[J].计算机应用,2008(1).

[10] Yu H,Gibbons P B,Nath S.Availability of multi-object operations[C]//Proceedings of NSDI'06,May 2006.

[11] 王梓坤,杨向群.生灭过程与马尔可夫链[M].北京:科学出版社,2005.

[12] Xin Q,Miller E,Schwarz T.Evaluation of distributed recovery in large-scale storage systems[C]//Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing,Honolulu,HI,June 4-6,2004.

[13] Yu Haifeng,Gibbons P B.Optimal inter-object correlation when replicating for availability[J].Distributed Computing,2009,21(5):367-384.

[14] 穆飞,薛巍.一种面向大规模副本存储系统的可靠性模型[J].计算机研究与发展,2009(5):756-761.

[15] Nath S,Yu H,Gibbons P B,et al.Subtleties in tolerating correlated failures in widearea storage systems[C]//Proceedings of NSDI'06,May 2006.

[16] van Renesse R.Efficient reliable Internet storage[C]//Proceedings of the Workshop on Dependable Distributed Data Management,Oct 2004.


(上接26页)


[7] 郝岩,冯象初,许建楼.一种非局部扩散的图像修复模型[J].西安电子科技大学学报,2010,37(5):825-828.

[8] Wu J Y,Ruan Q Q,An G Y.Exemplar-based image completion model employing PDE corrections[J].Informatica,2010,21(2):259-276.

[9] Xu Z B,Sun J.Image inpainting by patch propagation using patch sparsity[J].IEEE Trans Image Process,2010,19(5):1153-1165.

[10] Ji Hui,Shen Zuowei,Xu Yuhong.Wavelet frame based image restoration with missing/damaged pixels[J].East Asia Journal on Applied Mathematics,2011,1(2):108-131.

[11] Cai Jianfeng,Chan Raymond H,Shen Zuowei.Simultaneous cartoon and texture inpainting[J].Inverse Problems and Imaging,2010,4(3):379-395.

[12] Zhang Xiaoqun,Chan Tony F.Wavelet inpainting by nonlocal total variation[J].Inverse Problems and Imaging,2010,4(1):1-20.

[13] Yau Andy C,Tai Xuecheng.L0-norm and total variation inpainting[C]//Proceedings of SSVM 2009,2009,5567:539-551.

[14] Yau Andy C,Tai Xuecheng,Ng M.Compression and denoising using L0-norm[J].Computational Optimization and Applications,2011,50(2):425-444.

[15] 郝岩,冯象初,许建楼.交替迭代的变分修复模型及分裂Bregman算法[J].系统工程与电子技术,2011,33(12):2749-2754.

[16] Han Yu,Wang Weiwei,Feng Xiangchu.A new fast multiphase image segmentation algorithm based on nonconvex regularizer[J].Pattern Recognition,2012,45(1):363-372.

[17] 孙玉宝,费选,韦志辉,等.稀疏性正则化的图像泊松恢复模型及分裂Bregman迭代算法[J].自动化学报,2010,36(11):1512-1519.

[18] Hao Yan,Feng Xiangchu,Xu Jianlou.Multiplicative noise removal via sparse and redundant representations over learned dictionaries and total variation[J].Signal Processing,2012,92(6):1536-1549.

[19] Wang Z,Bovik A,Sheikh H,et al.Image quality assessment:from error visibility to structural similarity[J].IEEE Trans on Image Process,2004,13(4):1-14.