与副本结合的部分再生码

丁炳辰 李卫忠
(空军工程大学防空反导学院 西安 710051)

摘 要 (n,k,d)再生码允许存储节点传送所存数据的线性组合以及增加修复入度d,显著地降低了修复带宽,但是引入了更多的参与节点数及磁盘I/O。针对这一不足,提出了一种将复制方式与再生码结合的(n,k,d,λ,θ)部分再生码,并得到了与再生码类似的阈值函数和2个特殊点——最小存储量点和最小修复带宽点。部分再生码可以综合利用修复入度d和副本因子θ同时降低修复带宽和磁盘I/O。当所有的节点存储量相等时,部分再生码的单点修复带宽和磁盘I/O均优于再生码。定量比较的结果也显示,在最小存储量点,部分再生码比再生码有更低的平均修复带宽和平均磁盘I/O;在最小修复带宽点,部分再生码有更低的平均磁盘I/O以及与再生码相近的平均修复带宽。更重要的是,部分再生码适用于$d\leqslant n-2$的所有情形。

关键词 再生码,副本,修复带宽,磁盘I/O,修复入度
中图法分类号 TP302.8 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.040

Partially Regenerating Codes Combined with Replicas

DING Bing-chen LI Wei-zhong
(Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China)

Abstract (n,k,d) Regenerating codes(RC) significantly reduce repair bandwidth by allowing storage nodes to send the linear combinations of their data to the newcomer and increasing repair degree d. But they bring in more participating nodes and disk I/O. To solve this problem, this paper introduced partially regenerating codes(PRC) which combine RC (n,k,d,λ,θ) with replicas. PRC have also a threshold function and two special points: minimum-storage point and minimum-bandwidth point. PRC can simultaneously reduce repair bandwidth and disk I/O by utilizing repair degree d and replica factor θ. When the storage capacity of all nodes are the same, the repair bandwidth and disk I/O per node of PRC are superior to that of RC. The results of quantitative comparison also show that, comparing to RC, on minimum-storage point, PRC have less mean repair bandwidth and disk I/O; on minimum-bandwidth point, PRC have less mean disk I/O and mean repair bandwidth to similar RC. What's more important is that PRC is achievable when $d\leqslant n-2$.

Keywords Regenerating codes, Replica, Repair bandwidth, Disk I/O, Repair degree

1 引言

大规模分布式存储系统中数据量正在以指数的趋势增长,而系统中大量节点的暂时或永久性失效成为一种常态,要想可靠地存储这些数据,关键技术是引入冗余。相比简单的复制方式,纠删码(Erasure Codes)技术在使用相同的冗余量时提供了更高的可用性水平,或在相同的可用性水平下节省了大量的存储空间[1,2]。这一优点使得纠删码逐渐得到了大范围的应用,包括一些云存储系统(如新版本的GFS、Windows Azure[3])、大数据分析集群(如Facebook Analytics Hadoop Cluster[4])、归档存储系统和P2P存储系统。除了纠删码固有的编码和解码过程带来的运算能力的消耗,随后的研究还表明,传统的(n,k)纠删码,如RS编码,消耗了更多的网络带宽和磁盘I/O,通常情况下是复制方式的k倍[5,6]。这也是传统纠删码没有被当前的存储系统广泛应用的主要原因。

Dimakis等人[7]将网络编码(Network Coding)[8]应用到纠删码上,创造性地提出了(n,k,d)再生码(Regenerating Codes,RC)的概念,给出了单节点存储量与单节点修复带宽的最优折衷(Optimal Tradeoff)。再生码有效地降低了修复带宽,但是其降低修复带宽的方式——增大修复入度d,使得磁盘I/O的消耗在通常情况下比传统的纠删码更严重[6]。而随着新一代网络互联速度以及单个存储设备存储容量的增加,磁盘I/O正成为存储系统性能的主要瓶颈,并且修复入度的增加也使系统的抗扰动能力减弱。

针对再生码增加磁盘I/O的缺点,文献[9,10]提出了Repair-by-transfer/Uncoded repair的特殊编码方式,同时优化了修复过程的运算和磁盘I/O的消耗,但是只有在$d=n-1$时才能实现;文献[6]针对MSR编码进行优化,得到了同时满足"存储—可靠性"、"存储—修复带宽"以及"修复带宽—磁盘I/O"最优权衡的编码方式,但是其给出的编码也需要参数满足$d\geqslant 2k-2$;文献[11]通过特殊的设计降低了冗余节点的修复带宽和磁盘I/O,但是仅考虑了$d=n-1$的特殊情况;文

到稿日期:2016-01-08 返修日期:2016-03-01
丁炳辰(1992—),男,硕士生,主要研究方向为分布式存储系统、云计算、网络编码,E-mail:xiaoding16@aliyun.com;李卫忠(1968—),男,硕士,副教授,主要研究方向为云计算。

献[12]中的混合 MSR 编码降低了磁盘 I/O，但是当 $k\geqslant 5$ 时不存在线性精确修复编码。

上面提到的优化都依赖于再生码 $d>k$ 的特性，即当 d 等于或接近 k 时修复带宽和磁盘 I/O 均无法降低，而且至今也没有人提出一种适合所有参数的通用方式。本文针对单节点修复的情形，将复制方式与再生码结合，提出了部分再生码(Partially Regenerating Codes，PRC)，其适用于除 $d=n-1$ 以外所有参数的情形，并且有增加副本因子 θ 与增加 d 两种方式来降低修复带宽与磁盘 I/O，即使 $d=k$ 时，修复带宽和磁盘 I/O 也可以通过调节副本因子 θ 进行优化。通过综合利用 θ 和 d，甚至得到比 $d=n-1$ 时的再生码更优的修复带宽，却有更低的磁盘 I/O。

2 再生码 RC

文献[7]中，Dimakis 等人使用信息流图(Information Flow Graph)对分布式存储系统进行建模，描述了数据通过网络进行传递并最终汇集到数据重构节点的过程。通过对最小割的分析，得到了单节点存储量 α 与单节点修复带宽 γ 的最优折衷，达到这个最优折衷的纠删码称为(n,k,d)再生码。再生码以下列方式引入冗余：原始数据被分成大小相等的 k 个数据块，经过编码得到 $n(n>k)$个数据块，将这 n 个数据块放置在不同的存储节点上。n 个数据块中的任意 k 个足以重构出原始数据，所以最多能容忍$(n-k)$个节点的失效而不丢失原始数据。当有一个节点失效时，一个新的节点从剩下的有效节点中选择 $d(k\leqslant d<n)$个下载数据，然后利用下载的数据得到与失去的数据相同或功能上相同(保证任意 k 个足够重构出原始数据)的数据并将其存储。

最优折衷曲线上有 2 个特殊的点。

最小存储量点：

$$(\alpha_{MSR},\gamma_{MSR})=(\frac{B}{k},\frac{Bd}{k(d-k+1)})$$

最小修复带宽点：

$$(\alpha_{MBR},\gamma_{MBR})=(\frac{2Bd}{k(2d-k+1)},\frac{2Bd}{k(2d-k+1)})$$

相应的编码方案分别称为 MSR 编码和 MBR 编码。

3 部分再生码 PRC

3.1 编码策略及参数

将所有的存储节点分为两类：编码修复节点$\{X^i\}$和副本修复节点$\{Y^j\}$。对大小为 B 的原始数据编码后，放入$(n-1+\lambda)$个存储节点，其中有$(n-1)$个编码修复节点和副本数为 λ 的副本修复节点，即$\{X^i\}$有$(n-1)$个元素，$\{Y^j\}$有 λ 个元素；每个 X^i 存储数据大小为 α，每个 Y^j 存储数据大小为 θB(称 θ 为副本因子)，$\theta\in[0,1]$；原始数据利用 1 个 Y^j 和任意$(k-1)$个 X^i(共计 k 个存储节点)重构；当一个 X^i 失效时，一个新的 $X^{i'}$(称作新节点)仅从集合$\{X^i\}$中选择 $d(d\leqslant n-2)$个有效的存储节点(称作帮助节点)下载数据，从每个节点下载的数据大小为 $\beta(\beta\leqslant\alpha)$，修复带宽 $\gamma=d\beta(\alpha\leqslant\gamma)$；当一个 Y^j 失效时，一个新的 $Y^{j'}$ 从$\{Y^j\}$中选择一个有效的存储节点下载数据，修复带宽为 θB。n、λ、k、d 均是非负整数，α、β、γ 均是非负实数。

对每一个七元组$(n,k,d,\alpha,\gamma,\lambda,\theta)$，编码方案用$(n,k,d,\lambda,\theta)$表示，$C(\alpha)$表示所有的信息流图的最小割容量的下界，$\alpha^*(n,k,d,\gamma,\theta)$表示 α 的阈值函数。

图 1 给出了一个$(4,2,2,2,\alpha/B)$PRC 的信息流图，采用文献[7]中信息流图的表示方法：S 表示数据源节点，存储节点用输入点 in、输出点 out 和一条有向边 $in\to out$ 表示，数据汇集点 DC_i 可能有多个。S 将编码后的数据块传给 5 个节点：X^1，X^2，X^3，Y^1，Y^2，当 X^3 失效时，一个新的节点 X^4 利用 X^1 和 X^2 进行修复，DC 利用 Y^1 和 X^4 重构出原数据。

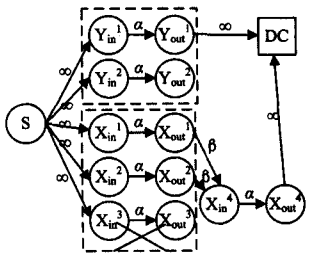


图 1 $(4,2,2,2,\alpha/B)$PRC 的一个信息流图

3.2 折衷曲线

定理 1 对任意的 $\alpha\geqslant\alpha^*(n,k,d,\gamma,\theta)$，七元组$(n,k,d,\alpha,\gamma,\lambda,\theta)$都是可行的，且线性网络编码可以满足下界。理论上，当 $\alpha<\alpha^*(n,k,d,\gamma,\theta)$时，$(n,k,d,\alpha,\gamma,\lambda,\theta)$是不可行的。阈值函数 $\alpha^*(n,k,d,\gamma,\theta)$如下：

$$\alpha^*(n,k,d,\gamma,\theta)=$$

$$\begin{cases}\dfrac{(1-\theta)B}{k-1}, & \gamma\in[f(0),+\infty)\\ \dfrac{(1-\theta)B-g(i)\gamma}{k-1-i}, & \gamma\in[f(i),f(i-1))\end{cases},i=1,\cdots,k-2 \quad (1)$$

其中，

$$f(i)\triangleq\frac{2(1-\theta)Bd}{(2k-i-3)i+2(k-1)(d-k+2)} \quad (2)$$

$$g(i)\triangleq\frac{(2d-2k+i+3)i}{2d} \quad (3)$$

证明：详细证明过程可以参考文献[7]附录中(n,k,d,α,γ)的 $\alpha^*(n,k,d,\gamma)$的求解过程，下面仅列出关键计算结果：

$$b_i\triangleq(1-\frac{k-2-i}{d})\gamma=[d-(k-2)+i]\beta,i=0,1,\cdots,k-2$$

$C(\alpha)=\theta B+\sum_{i=0}^{k-2}\min\{b_i,\alpha\}(\alpha\in[0,b_{k-2}])$是一个分段线性函数：

$$C(\alpha)=\begin{cases}\theta B+(k-1)\alpha, & \alpha\in[0,b_0]\\ \theta B+b_0+(k-2)\alpha, & \alpha\in(b_0,b_1]\\ \vdots & \vdots\\ \theta B+b_0+\cdots+b_{k-3}+\alpha, & \alpha\in(b_{k-3},b_{k-2}]\end{cases}$$

满足 $C(\alpha)\geqslant B$ 的 α 的最小值为：

$$\alpha^*=\begin{cases}\dfrac{(1-\theta)B}{k-1}, & B\in[\theta B,\theta B+(k-1)b_0]\\ \dfrac{(1-\theta)B-\sum_{j=0}^{i-1}b_j}{k-1-i}, & B\in(\theta B+\sum_{j=0}^{i-1}b_j+(k-i-1)b_{i-1},\\ & \quad \theta B+\sum_{j=0}^{i}b_j+(k-i-2)b_i]\end{cases}$$

$i=1,\cdots,k-2$

其中，

$$\sum_{j=0}^{i-1}b_j=\sum_{j=0}^{i-1}(1-\frac{k-2-j}{d})\gamma=\gamma i[\frac{d-k+2}{d}+\frac{i-1}{2d}]=\gamma g(i)$$

$$\sum_{j=0}^{i} b_j + (k-i-2)b_i$$
$$= \sum_{j=0}^{i}(1-\frac{k-2-i}{d})\gamma + (k-i-2)(1-\frac{k-2-i}{d})\gamma$$
$$= \gamma(i+1)[\frac{d-k+2}{d}+\frac{i}{2d}] + (k-i-2)(\frac{d-k+2}{d}+\frac{i}{d})\gamma$$
$$= \gamma\frac{(k-1)(d-k+2)}{d} + \gamma\frac{i(2k-i-3)}{2d}$$
$$= \gamma\frac{(1-\theta)B}{f(i)}$$

因此，有

$$\alpha^* = \frac{(1-\theta)B - g(i)\gamma}{k-1-i}$$

$$B \in (\theta B + \gamma\frac{(1-\theta)B}{f(i-1)}, \theta B + \gamma\frac{(1-\theta)B}{f(i)}], i=1,\cdots,k-2$$

所以有式(1)成立。证毕。

同 RC 一样，PRC 也有 2 个特殊点。

最小存储量点：

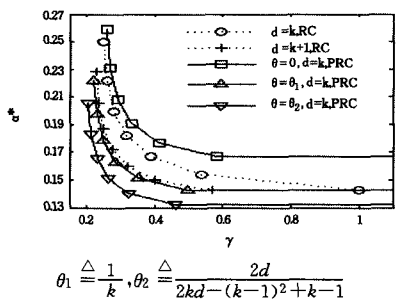
$$(\alpha_{PMSR}, \gamma_{PMSR}) = (\frac{(1-\theta)B}{k-1}, \frac{(1-\theta)Bd}{(k-1)(d-k+2)})$$

最小修复带宽点：

$$(\alpha_{PMBR}, \gamma_{PMBR}) = (\frac{2(1-\theta)Bd}{(k-1)(2d-k+2)}, \frac{2(1-\theta)Bd}{(k-1)(2d-k+2)})$$

实现这两个点的编码分别称为 PMSR 编码和 PMBR 编码。注意到 PMBR 与 MBR 相似，有 $\alpha_{PMBR} = \gamma_{PMBR}$。

图 2 给出了 $k=7$ 时 RC 与 PRC 的(γ, α^*)折衷曲线，数据在 $B=1$ 下计算得到。可以看到，随着 θ 的增大，PRC 的折衷曲线逐渐下降，类似于 RC 增大 d 的效果，但是 d 的增大不仅使得磁盘 I/O 上升，也使得系统的抗扰动能力减弱。而在下一节我们将看到，θ 的增大反而会降低磁盘 I/O。折衷曲线的降低意味着 α 的减小，除了最小存储量点以外，读取原文件时网络传输的数据量以及磁盘 I/O 即$(k-1)\alpha+\theta B$ 均减小了，对于大文件且读取频繁的情况，带宽和磁盘 I/O 的降低是不可忽视的。



$$\theta_1 \triangleq \frac{1}{k}, \theta_2 \triangleq \frac{2d}{2kd-(k-1)^2+k-1}$$

图 2　$k=7$ 时 RC 与 PRC 的(γ, α^*)折衷曲线

图 2 中 θ 取了 3 个特殊值：0，$\theta_1 \triangleq \frac{1}{k}$，$\theta_2 \triangleq \frac{2d}{2kd-(k-1)^2+k-1}$。当$\theta=0$ 时，即为$(n-1, k-1)$RC；当 $\theta=\theta_1$ 时，$\theta B = \alpha_{PMSR} = B/k$，此时 PMSR 编码的所有节点存储量相等；当 $\theta=\theta_2$ 时，$\theta B = \alpha_{PMBR}$，此时 PMBR 编码的所有节点存储量相等。注意 $\theta_2 \geqslant \theta_1$。另外，当 $\theta=1$ 时，即为副本数为 λ 的简单复制方式。

通过代数变换，易得下列性质：

性质 1

a) $\alpha^*(n,k,d+1,\gamma,\theta) \leqslant \alpha^*(n,k,d,\gamma,\theta)$；

b) $\alpha^*(n,k,d,\gamma,\theta') < \alpha^*(n,k,d,\gamma,\theta'), 0 \leqslant \theta' < \theta' \leqslant 1$；

c) $\alpha_{PMSR}, \gamma_{PMSR}, \alpha_{PMBR}, \gamma_{PMBR}$ 均随着 θ 的增大而减小；

d) $\gamma_{PMSR}, \alpha_{PMBR}, \gamma_{PMBR}$ 均随着 d 的增大而减小；

e) $\alpha_{PMSR} \leqslant \alpha_{MSR} \Leftrightarrow \theta \geqslant \theta_1$；

f) $\gamma_{PMSR} \leqslant \gamma_{MSR} \Leftrightarrow \theta \geqslant \theta_1 - \frac{k-1}{k(d-k+1)}$；

g) $\alpha_{PMBR} \leqslant \alpha_{MBR} \Leftrightarrow \theta \geqslant \theta_1 - \frac{k-1}{k(2d-k+1)}$。

4　评估

4.1　假设、评估模型及评估量

做以下一些简化的假设。

假设有大量动态的节点协作地存储数据，节点可用性是独立同分布(I. I. D.)[5]的，并且可用的节点数不变。

假设数据的可用性不变，即某个节点失效后，立即进行修复，而不是采取延迟修复[13]，导致一定时间内数据可用性的降低。

假设数据是不变的，因此在任意时刻，3.1 小节所述的数据重构和节点修复都是可行的，排除了由数据更新所带来的带宽与磁盘 I/O 消耗。在许多云存储系统，如 Windows Azure、HDFS[14]，大量数据都是不变的。

假设 X^i 的修复过程中，d 个帮助节点读取所有存储的数据 α，计算其线性组合得到 β。除去一些磁盘 I/O 特殊优化的编码[6,9,10]，传统的修复模式就是这样。

在文献[7]中的评估模型基础上稍作修改，得到如下评估模型：

1)每单位时间，$\{X^i\}$中永久性失效的节点比例是 f_1，$\{Y^j\}$中永久性失效的节点比例是 f_2，且 $0<f_1<1, 0<f_2<1$；

2)在任意时间点，X^i 可用性是 a_1，Y^j 可用性是 a_2，且 $0<a_1<1, 0<a_2<1$。

表 1 列出了(n,k,d,λ,θ)PRC 与(n,k,d)RC 的评估量，修复带宽和磁盘 I/O 均有单节点和均值两种评估量。其中 $S_{MSR} = n\alpha_{MSR}$，$S_{MBR} = n\alpha_{MBR}$，$S_{PMSR} = (n-1)\alpha_{PMSR} + \lambda\theta B$，$S_{PMBR} = (n-1)\alpha_{PMBR} + \lambda\theta B$。

$$P_{RC} = \sum_{i=0}^{k-1}\binom{n}{i}a_1^i(1-a_1)^{n-i} \quad (4)$$

$$P_{PRC} = 1-b+b*\sum_{i=0}^{k-2}\binom{n-1}{i}a_1^i(1-a_1)^{n-1-i}, \theta \neq 0,1 \quad (5)$$

$b \triangleq 1-(1-a_2)^\lambda$ 表示有 λ 个元素的$\{Y^j\}$形成的整体的可用性。

表 1　本文使用的评估量

编码		总存储量	修复带宽			磁盘 I/O			原数据的不可用性
			X^i	Y^j	均值	X^i	Y^j	均值	
RC	MSR	S_{MSR}	γ_{MSR}	\	$f_1 n\gamma_{MSR}$	$d\alpha_{MSR}$	\	$f_1 nd\alpha_{MSR}$	P_{RC}
	MBR	S_{MBR}	γ_{MBR}	\	$f_1 n\gamma_{MBR}$	$d\alpha_{MBR}$	\	$f_1 nd\alpha_{MBR}$	
PRC	PMSR	S_{PMSR}	γ_{PMSR}	θB	$f_1(n-1)\gamma_{PMSR}+f_2\lambda\theta B$	$d\alpha_{PMSR}$	θB	$f_1(n-1)d\alpha_{PMBR}+f_2\lambda\theta B$	P_{PRC}
	PMBR	S_{PMBR}	γ_{PMBR}	θB	$f_1(n-1)\gamma_{PMBR}+f_2\lambda\theta B$	$d\alpha_{PMBR}$	θB	$f_1(n-1)d\alpha_{PMSR}+f_2\lambda\theta B$	

4.2 单节点修复带宽及磁盘 I/O

对于最小存储量点，PMSR 在 $\theta=\theta_1$ 时，$\alpha_{PMSR}=\theta B=\alpha_{MSR}$，$X^i$ 的单点磁盘 I/O 与 MSR 的单点磁盘 I/O 相等，Y^j 的单点磁盘 I/O 是 MSR 的 $\frac{1}{d}$；由性质 1 中 f)可知 $\gamma_{PMSR}<\gamma_{MSR}$，降低比例为 $\frac{1}{d-k+2}$，当 $d=k$ 时有最大的降低比例 $\frac{1}{2}$，且 Y^j 的单点修复带宽小于 MSR 的单点修复带宽，降低比例为 $\frac{k-1}{d}$，当 $d=k$ 时有最大的降低比例 $(1-\frac{1}{k})$。

对于最小修复带宽点，PMBR 在 $\theta=\theta_2$ 时，$\theta B=\alpha_{PMBR}$，再由性质 1 中 g)可知 $\alpha_{PMBR}<\alpha_{MBR}$，因此 X^i，Y^j 的单点磁盘 I/O 均小于 MBR 的；由于 $\alpha_{MBR}=\gamma_{MBR}$，$\alpha_{PMBR}=\gamma_{PMBR}$，因此 X^i，Y^j 的单点修复带宽均小于 MBR 的。

4.3 定量比较

首先给出引理 1 与定理 2，它们描述了 (n,k)RC 中 n,k 与平均节点可用性 a 的关系，这对理解后面的仿真结果有帮助。

引理 1 在 4.1 节中假设条件下，对于 (n,k)RC，n,k 的取值必须依赖于存储数据所期望的不可用性 ε（即 $1-\varepsilon$ 有多少个 9）和平均节点可用性 a，则有下列等式成立：

$$1-\varepsilon=\sum_{i=k}^{n}\binom{n}{i}a^i(1-a)^{n-i} \tag{6}$$

$$\frac{n}{k}=\left[\frac{\sigma_\varepsilon\sqrt{\frac{a(1-a)}{k}}+\sqrt{\frac{\sigma_\varepsilon^2 a(1-a)}{k}+4a}}{2a}\right]^2 \tag{7}$$

其中，σ_ε 是对应于期望的可用性水平 $(1-\varepsilon)$ 的正态分布的标准差。例如 $\sigma_\varepsilon=3.7$ 对应于 4 个 9 的可用性，即 0.9999。

证明：因为数据可用性是至少 k 个节点可用的概率，所以有式(6)成立，利用代数化简和二项分布的正态近似，由式(6)可推得式(7)，完整表述和推导过程见文献[5,15]。证毕。

定理 2 在 4.1 节中假设条件下，对于满足存储数据可用性 $1-\varepsilon$ 的 $(n-m,k-m)$RC，随着平均节点可用性 a 的降低，n 逐渐增大，且 m 越大，n 增长的幅度越小。

证明：由引理 1 可得，

$$n=\left[\frac{\sigma_\varepsilon\sqrt{\frac{a(1-a)}{k-m}}+\sqrt{\frac{\sigma_\varepsilon^2 a(1-a)}{k-m}+4a}}{2a}\right]^2*(k-m)+m$$

$$=\frac{1}{4}[z(a)]^2+m$$

$$z(a)\triangleq\sigma_\varepsilon\sqrt{\frac{(1-a)}{a}}+\sqrt{\frac{\sigma_\varepsilon^2(1-a)}{a}+\frac{4(k-m)}{a}}$$

当 a 减小时，$z(a)$ 增大，且 m 越大，$z(a)$ 增长的幅度越小。也即 a 减小时，n 增大，且 m 越大，n 增长的幅度越小。证毕。

本节中取 $1-\varepsilon=0.9999$，这是故障可恢复系统的可用性水平，即 1 年有 1 小时的停机时间[16]。更高的数据可用性是不必要的，不仅因为需要更高的开销，而且系统其他没有达到这个可用性水平的因素，如链路等，将会制约整个系统的可用性[5]。仍使用文献[7]中的数据，由于是比较 PRC 与 RC 的相对性能，这样做并不会影响结论。表 2 列出了该数据。

表 2 本文使用的可用性

平台	时长（天）	开始日期	平均在线节点数	f（每天的失效节点比例）	a
PlanetLab	527	Jan. 2004	303	0.017	0.97
Microsoft PCs	35	July 6,1999	41970	0.038	0.91
Skype	25	Sept. 12,2005	710	0.12	0.65
Gnutella	2.5	May 2001	1846	0.30	0.38

图 3 给出了 $k=7$ 时定量比较的结果，数据在 $B=1$GB 下计算得到。

图 3 中，(a)、(b)、(c)、(d)中每个坐标系的左 Y 轴表示 (n,k,d,λ,θ)PRC 相对 (n,k,d)RC 在每个评估量上的减少比例，对应于图中的实线；右 Y 轴表示 Y^j 与 X^i 存储量之比：$\frac{\theta B}{\alpha}$，对应于图中的虚线，图中仅显示出了虚线对应的 $\frac{\theta B}{\alpha}\in[0,10]$的部分。线上标记的点对应于 $\theta B=\alpha$ 的特殊情形。图 3(e)、图 3(f)给出了 PlanetLab、Microsoft PCs 平台下 $\theta B=\alpha$ 时，随着 d 的增大，PRC 与 RC 的平均修复带宽与平均磁盘 I/O 的变化。注意到 RC 中 $d\leqslant n-1$，而 PRC 中 $d\leqslant n-2$。

由于 MBR 与 PMBR 均有 $\alpha=\gamma$，根据后面的 f_1、f_2、a_1、a_2 的设置，总存储与平均修复带宽是成比例的，所以增大或减小的比例是相等的，因此图 3 省去了最小修复带宽点的总存储曲线；并且，由于 d 并不改变 $\theta B=\alpha$ 时 PMSR 与 MSR 的存储量（$\theta B=\alpha_{PMSR}=\alpha_{MSR}=\frac{B}{k}$），因此图 3(e)、图 3(f)中也省去了最小存储量点的总存储曲线。

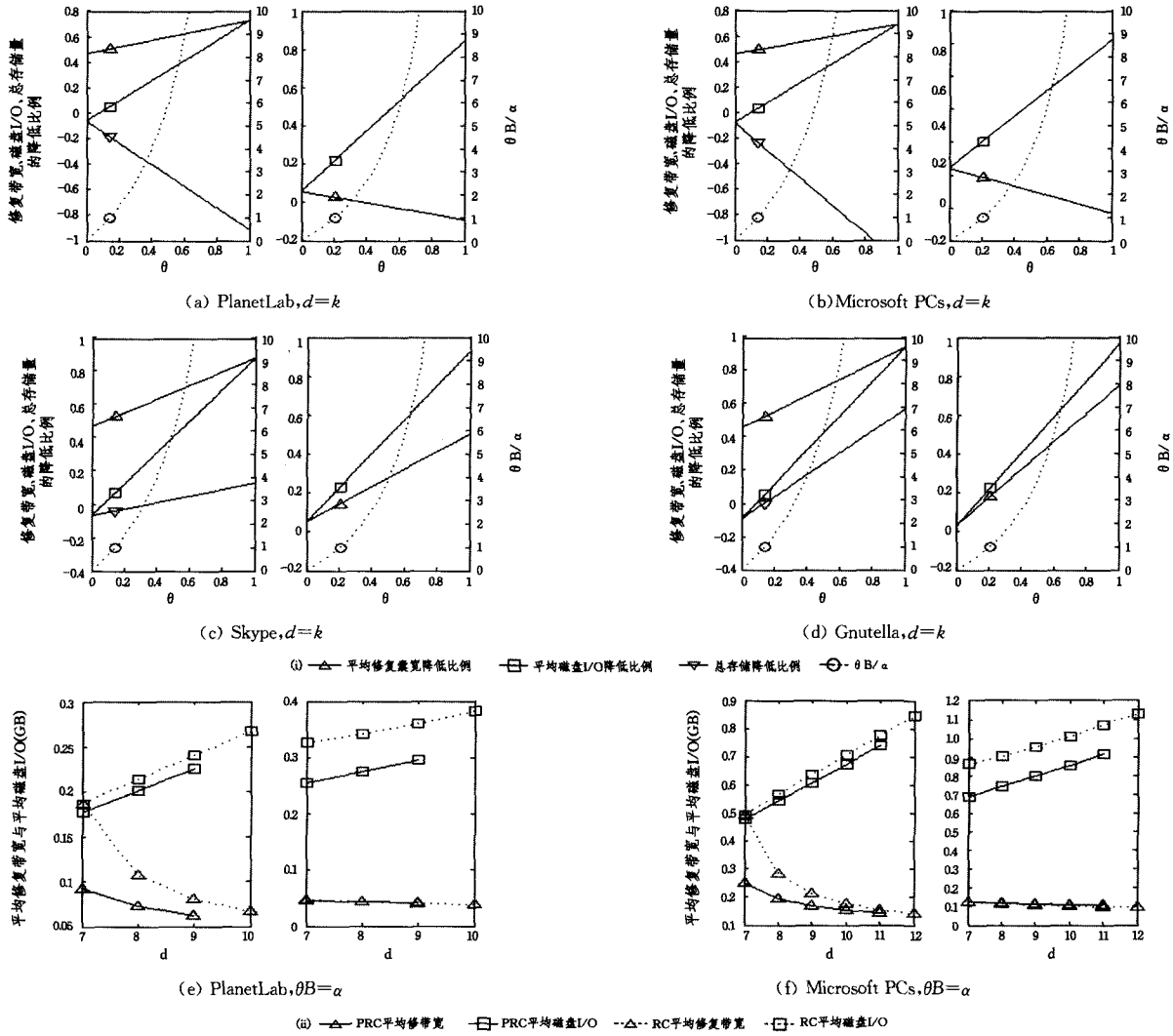
注意：虽然式(5)的成立要求 $\theta\neq0,1$，但是图 3 中 $\theta=0$ 和 $\theta=1$ 分别真实反映了 $(n-1,k-1)$RC 和简单复制方式的情形。

对于节点可用性较高的平台，如 PlanetLab、Microsoft PCs，取 $a_1=a_2$，$f_1=f_2$，即 X^i 与 Y^j 可用性相同。对于节点可用性较低的平台，如 Skype、Gnutella，为了避免 λ 的值过大带来的存储消耗，引入高可用性节点作为 Y^j，a_2 均取 PlanetLab 的 a 值，虽然从理论上有 $f_2<f_1$，不过为了保持 PMBR 总存储与平均修复带宽成比例的特性，仿真中仍然取 $f_2=f_1$，这样会使得结果偏小，不过并不影响结论。

为了满足 4 个 9 的数据可用性，PlanetLab 平台下 RC 与 PRC 的 n 值均为 11，$\lambda=3$；Microsoft PCs 平台下 RC 与 PRC 的 n 值均为 13，$\lambda=4$；Skype 平台下 RC 与 PRC 的 n 值分别为 24，23，$\lambda=3$；Gnutella 平台下 RC 与 PRC 的 n 值分别为 48，46，$\lambda=3$。n 值变化的原因如下：令 $Q\triangleq\sum_{i=0}^{k-2}\binom{n-1}{i}a_1^i(1-a_1)^{n-1-i}$，则 $P_{PRC}=Q+(1-Q)(1-b)$。当 Q 与 $(1-b)$ 相差 1 个数量级时，如上述仿真中为 10^{-4} 与 10^{-5}，$P_{PRC}\approx Q$。注意到 Q 是 $(n-1,k-1)$RC 的数据不可用性，P_{RC} 是 (n,k)RC 的不可用性，根据定理 2，随着平台节点可用性的降低，为保证 4 个 9 的数据可用性，二者的 n 均会增大，并且后者增大的幅度更大。

从图 3(a)、(b)中可以看出，随着 θ 的增大，在 PlanetLab 和 Microsoft PCs 平台上，PMSR 会损失存储，但是修复带宽和磁盘 I/O 均有较大降低，尽管有 $d=k$，但在损失少量存储时，尤其是修复带宽仍有较大降低，如特殊点 $\theta B=\alpha$ 的情形：在 PlanetLab 平台上损失 18%存储，降低 51%的修复带宽；在 Microsoft PCs 平台上损失 23%的存储，降低 49%的修复带

宽。PMBR 在保持存储与修复带宽少量增大或减小时，磁盘 I/O 有较大降低，如特殊点 $\theta B=\alpha$ 的情形：在 PlanetLab 平台上存储与修复带宽均降低 3%，磁盘 I/O 降低 22%；在 Microsoft PCs 平台上损失 1%的存储与修复带宽，磁盘 I/O 降低了 20%。Skype、Gnutella 平台中由于引入了高可用性节点作为 Y^j，并且 n 较大，因此随着 θ 的增大，Y^j 带来的修复带宽、总存储与磁盘 I/O 的增加被 X^i 带来的降低所掩盖，整体呈现出 θ 越大而修复带宽、总存储与磁盘 I/O 越小的趋势。



注：(i)中的实例适用于(a)、(b)、(c)、(d)，(ii)中的实例适用于(e)、(f)。

图 3　$k=7$ 时 PRC 与 RC 的性能对比

由性质 1 可知，PRC 也可以通过增大 d 来降低修复带宽，同 RC 一样，d 的增大会带来磁盘 I/O 的增加，这与图 3(e)、(f)显示的结果是一致的。对于 PMBR 与 MBR，d 的增大所带来的修复带宽的降低是不明显的，却增大了磁盘 I/O，但 PMBR 始终保持比 MBR 更低的磁盘 I/O。对于 PMSR 与 MSR，d 的增大对修复带宽的降低是明显的，但是，我们可以看到 PMSR 在 $d=k$ 时的带宽已经接近 MSR 在 $d=n-1$ 时的带宽，甚至 PMSR 在 $d=n-2$ 时的带宽比 MSR 在 $d=n-1$ 时更低，并且伴随着更低的磁盘 I/O。

结束语　本文将复制方式与再生码结合，提出了一种新的编码方案：(n,k,d,λ,θ)部分再生码。部分再生码可以综合利用修复入度 d 与副本因子 θ 来优化总存储、修复带宽和磁盘 I/O。与(n,k,d)再生码相比，在修复入度相等时，PMSR 比 MSR 有更优的修复带宽与磁盘 I/O，PMBR 比 MBR 有更优的磁盘 I/O 和相近的修复带宽。由于部分再生码引入了复制方式，因此复制方式固有的存储量损失的缺陷也引入到了部分再生码中，在 PMSR 中较为明显，但是对于多数分布式存储系统而言，带宽与磁盘 I/O 是制约系统性能的主要瓶颈[5,6]。另外，由于部分再生码的编码修复部分仍然是再生码的结构，因此关于再生码的一些优化，如精确修复[17,18]、多节点协作修复[19,20]、磁盘 I/O[6]等，均可以应用其中。

参 考 文 献

[1] Weatherspoon H, Kubiatowicz J D. Erasure coding vs. replication: A quantitative comparison[C]// International Workshop on Peer-to-Peer Systems. Cambridge, Massachusetts, United States: Springer Berlin Heidelberg, 2002: 328-337

[2] Luo Xiang-hong, Shu Ji-wu. Summary of Research for Erasurce Code in Storage System[J]. Journal of Computer Research and Development, 2012, 49(1): 1-11(in Chinese)
罗象宏，舒继武. 存储系统中的纠删码研究综述[J]. 计算机研究与发展，2012，49(1)：1-11

[3] Huang Cheng, Simitci H, Xu Yi-kang, et al. Erasure coding in windows azure storage[C]// Usenix Conference on Technical Conference. Berkeley, CA, USA: USENIX, 2012: 2

[4] Sathiamoorthy M, Asteris M, Papailiopoulos D, et al. XORing elephants: Novel erasure codes for big data[C]// VLDB Endowment. Riva del Garda, Italy: dbTrento, 2013: 325-336

[5] Rodrigues R, Liskov B. High availability in DHTs: Erasure coding vs. replication[C]// Proc of IPTPS. Ithaca, NY, USA: Springer Berlin Heidelberg, 2005: 226-239

[6] Rashmi K V, Nakkiran P, Wang Jing-yan, et al. Having Your Cake and Eating It Too: Jointly Optimal Erasure Codes for I/O, Srorage, and Network-banwidth[C]// USENIX Conference on File and Storage Technologies. Santa Clare, CA, USA: USENIX, 2015: 81-94

[7] Dimakis A G, Godfrey P B, Wu Yun-nan, et al. Network coding for distributed storage systems[J]. IEEE Trans on Information Theory, 2010, 56(9): 4539-4551

[8] Ahlswede R, Cai N, Li S Y R, et al. Network information flow [J]. IEEE Trans on Information Theory, 2000, 46(4): 1204-1216

[9] Lin S J, Chung W H. Novel Repair-by-Transfer Codes and Systematic Exact-MBR Codes with Lower Complexities and Smaller Field Sizes[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(12): 3232-3241

[10] Hu Yu-chong, Lee P, Shum K W. Analysis and construction of functional regenerating codes with uncoded repair for distributed storage systems[C]// INFOCOM. Turin: IEEE, 2013: 2355-2363

[11] Yang Bin, Tang Xiao-hu, Li Jie. A Systematic Piggybacking Design for Minimum Storage Regenerating Codes[J]. IEEE Transactions on Information Theory, 2015, 61(11): 5779-5786

[12] Liang Song-tao, Liang Wen-juan, Kan Hai-bin. Construction of one special minimum storage regenerating code when $\alpha=2$[J]. Science China Information Sciences, 2015, 58(8): 1-10

[13] Bhagwan R, Tati K, Cheng Yu-chung, et al. Total recall: System support for automated availability management[J]. Nsdi, 2004, 1: 337-350

[14] Shvachko K, Kuang H, Radia S, et al. The Hadoop distributed file system[C]// Symp. on Mass Storage Systems and Technologies. USA: IEEE, 2010: 1-10

[15] Bhagwan R, Savage S, Voelker G M. Understanding availability [C]// International Workshop on Peer-to-Peer Systems. Berkeley, California, USA: Springer Berlin Heidelberg, 2003: 256-257

[16] Hwang K, Fox G C, Dongarra J J. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things[M]. Morgan Kaufmann, 2011

[17] Dimakis A G, Ramchandran K, Wu Yun-nan, et al. A Survey on network codes for distributed storage[J]. Proceeding of the IEEE, 2011, 99(3): 476-489

[18] Rashmi K V, Shah N B, Kumar P V. Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction[J]. IEEE Transaction on Information Theory, 2011, 57(8): 5227-5239

[19] Hu Yu-chong, Xu Yin-long, Wang Xiao-zhao, et al. Cooperative Recovery of Distributed Storage Systems from Multiple Losses with Network Coding[J]. IEEE Journal on Selected Areas in Communications, 2010, 28(2): 268-276

[20] Kermarrec A M, Scouarnec N L, Straub G. Repairing multiple failures with coordinated and adaptive regenerating codes[C]// International Symposium on Network Coding. Beijing: IEEE, 2011: 1-6

(上接第 183 页)

温昱晖,陈广勇,赵劲涛. 基于 CP-ABE 在云计算中实现数据访问控制的方案[J]. 重庆邮电大学学报(自然科学版), 2013, 25(5): 658-664

[8] Chase M. Multi-Authority attribute based encryption[C]// Proc of the 4th Theory of Cryptography Conf. Germany: Springer Berlin Heidelberg, 2007: 515-534

[9] Ruj S, Nayak A, Stojmenovic I. DACC: Distributed access control in clouds[C]// Proc of the 10th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications. Washington, DC: IEEE Press, 2011: 91-98

[10] Liu Xue-jiao, Xia Ying-jie, Jiang Sha-sha, et al. Hierarchical attribute-based access control with authentication for outsourced data in cloud computing[C]// Proc of the 2013 12th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications. Australia: IEEE Press, 2013: 477-484

[11] Yang Geng, Wang Dong-yang, Zhang Ting, et al. Attribute-Based Access Control with Multi-Authority Structure in Cloud Computing[J]. Journal of Nanjing University of Posts and Telecommunications(Natural Science), 2014, 34(2): 2-9(in Chinese)
杨庚,王东阳,张婷,等. 云计算环境中基于属性的多权威访问控制方法[J]. 南京邮电大学学报(自然科学), 2014, 34(2): 2-9

[12] Huang Xiao-fneg, Qi Tao, Qin Bao-dong, et al. Multi-Authority Attribute Based Encryption Scheme Revocation[C]// 2015 24th International Conference on Computer Communication and Networks (ICCCN). IEEE Press, 2015: 1-5

[13] Chen Yan-li, Song Ling-ling, Yang Geng. Attribute-based access control for multi-authority system with constant size ciphertext in cloud computing[J]. Wireless Communication Over Zigbee for Automotive Inclination Measurement China Communications, 2016, 13(2): 146-162

[14] Xu X, Zhou J, Wang X, et al. Multi-Authority proxy re-encryption based on CPABE for cloud storage systems[J]. Journal of Systems Engineering and Electronics, 2016, 27(1): 211-223

[15] Chen Dan-wei, Wan Liang-qing, Wang Chen, et al. A Multi-authority Attribute-Based Encryption Scheme with Pre-decryption [C]// 2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP). IEEE Press, 2015: 223-228

[16] Li Yong, Zeng Zhen-yu, Zhang Xiao-fei. Outsourced decryption scheme supporting attribute revocation[J]. Journal of Tsinghua University(Sci & Technol), 2013, 53(12): 1664-1669(in Chinese)
李勇,曾振宇,张晓菲. 支持属性撤销的外包解密方案[J]. 清华大学学报(自然科学版), 2013, 53(12): 1664-1669

[17] Yang Kan, Jia Xiao-hua, Ren Kui, et al. DAC-MACS: Effective data access control for multi-authority cloud storage systems [J]. IEEE Transactions on Information Forensics and Security, 2013, 8(11): 1790-1801

[18] Rong Xing, Zhao Yong, Jiang Rong. MMACS: A Multi-Authority Cloud Access Scheme with Mixed Access Structure[C]// Proc of Workshop on Secure Networking and Forensic Computing. Sydney, NSW: IEEE Press, 2014: 706-711