基于矩阵运算的最小冗余存储再生码MSRRC研究

王 禹[1,2] 赵跃龙[2] 侯 昉[3]

(广东技术师范学院教育技术与传播学院 广州 510665)[1]

(华南理工大学计算机科学与工程学院 广州 510640)[2] (广东金融学院计算机系 广州 510520)[3]

摘 要 分布式存储系统常常使用纠删码冗余技术提高数据的安全性和可靠性,从而使系统具有自修复失效数据的能力,但传统纠删码在修复失效节点时需要传输的数据量较大。再生码是纠删码的一种改进形式,它的主要特点是无需下载整个数据文件就能恢复单个节点数据,从而有效减少了数据修复时的网络带宽。相关文献证明数据修复时存在最小存储再生点(MSR),由此提出最小冗余存储再生码MSRRC。本研究主要采用数据矩阵和修复矩阵实现MSRRC再生码,通过实例详细给出再生码的实现过程,并理论证明其正确性,最后仿真实验验证了MSRRC的有效性。

关键词 分布式系统,再生码,数据修复

中图法分类号 TP316 **文献标识码** A

Minimum Redundancy Storage Regeneration Code Research MSRRC Based on Matrix Operation

WANG Yu[1,2] ZHAO Yue-long[2] HOU Fang[3]

(School of Education and Technology Guangdong Polytechnic Normal University, Guangzhou 510665, China)[1]

(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China)[2]

(Department of Computer, Guangdong University of Finance, Guangzhou 510520, China)[3]

Abstract Distributed storage systems often use erasure code redundancy technology to improve the safety and reliability of the data, so that the system has the ability to self repair failure data, but the traditional erasure codes need to transfer the large amount of data in order to repair failure nodes. Regeneration code is an improved form of erasure code, and its main characteristic is that it does not need to download the entire data file when restoring a single node data, which can effectively reduce the network bandwidth when repairing data. The relevant documents prove that data repair has minimal storage regeneration point(MSR), so presents minimum redundancy storage regeneration code MSRRC. Research mainly used the data matrix and the repair matrix to achieve MSRRC, and through examples, detailly introduced the realization process of regenerating codes, and proved the correctness of the theory. The simulation results verify the validity of MSRRC.

Keywords Distributed system, Regeneration code, Data repairing

为提高系统可靠性,分布式存储系统常常使用纠删码技术,这些系统有RAID-6[1],OceanStore[2]、Total Recall[3]等。假定存储到系统的数据文件由B个信息符号(Message Symbol)的集合表示,每个信息符号都属于大小为q的有限域F_q,(n,k)纠删码分块将分布存储到n个节点,汇聚节点(DC)连接任意的k个存活节点就能重构数据文件[4]。然而当某个节点失效时,系统必须具有自修复能力再生(regenerate)失效节点[5]。最直接的方法是置换节点从任意的k个存活节点下载纠删码编码数据,从而提取失效节点数据,这种方法将增大系统的计算以及网络和存储开销。因此,是否存在更好的冗余数据编码方法成为当前的研究热点[6-8],相关文献[9-11]提出了不同的网络编码技术。

1 相关研究工作

在文献[13]的冗余再生码方案中,数据文件编码存储到n个节点,每个节点存储大小为α的编码数据。当某个节点失效时,置换节点能从剩余$n-1$个节点连接任意d个节点($k\leqslant d\leqslant n-1$),从每个节点下载$\beta,\beta\leqslant\alpha$。帮助修复的$d$个节点称为帮助节点(Help Nodes),用于修复的整个数据下载量$d\beta$称为修复带宽(Repair Bandwidth)。冗余再生码方案的参数集可表示为$\{[n,k,d],(\alpha,\beta,B)\}$,其平均修复带宽$d\beta$小于文件大小$B$。各相关参数如图1所示,该图描述了数据重构和失效节点数据修复的过程[12]。

本文受国家自然科学基金(60573145),广东省科技创新项目(2013KJCX0116),广东省教育科学规划项目(2012JK048),广东高校优秀青年创新人才培养计划项目(2012WYM_0088),数字媒体本科专业核心课程体系研究项目资助。

王 禹(1974-),男,博士生,副教授,主要研究方向为计算机网络存储、计算机系统结构、计算机网络与通信,E-mail:wangyu_csu@163.com;赵跃龙(1958-),男,博士,教授,博士生导师,主要研究方向为计算机外存储系统;侯 昉(1975-),男,博士生,讲师,主要研究方向为计算机系统结构。

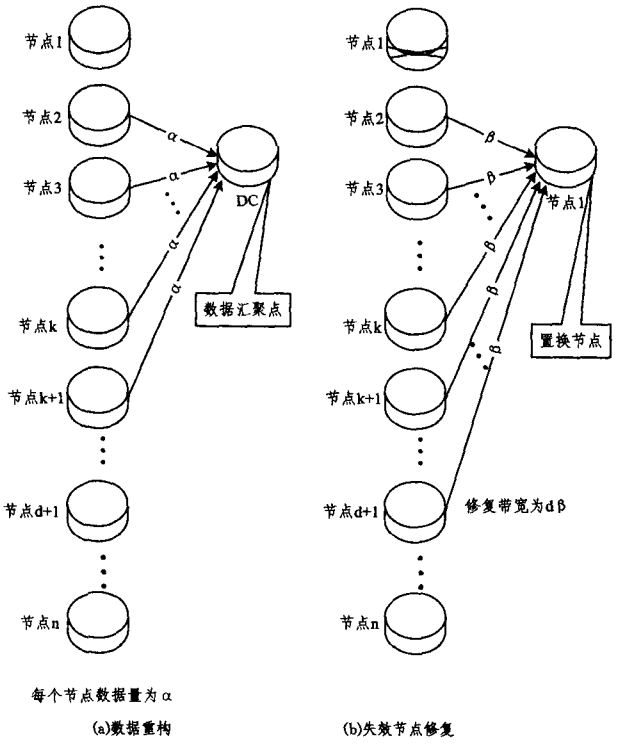

图1 冗余再生码执行过程

文献[14]提出了修复失效节点的最小带宽 Deterministic regenerating 方案(只给出了方案的理论性描述,没有给出其具体实现),方案中特别关注的是其平衡曲线中的极值点:最小带宽再生点(MBR)和最小存储再生点(MSR),如图2[15]所示。前者通过提高每个节点存储量来实现最小带宽,而后者则是每个节点存储尽可能少的数据量的同时最小化使用带宽。

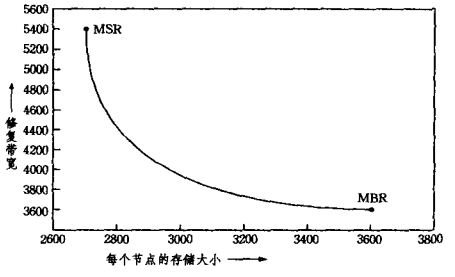

图2 存储-带宽平衡曲线:每个节点的存储大小 α 与修复带宽 $d\beta$

最小存储再生点(MSR)能从式(1)取得:

$$(\alpha_{MSR},\beta_{MSR})=\left(\frac{B}{k},\frac{Bd}{k(d-k+1)}\right) \tag{1}$$

式中,B 指源文件的大小,α 指节点的存储开销,β 为失效修复的通信开销,d 指存活节点个数,k 指数据重构所需的节点个数。

文献[16]证明了冗余再生码必须满足的条件:

$$\sum_{i=0}^{k-1}\min\{\alpha,(d-i)\beta\}\geqslant B \tag{2}$$

在实现最小化 α 和 β 的过程中,最小化 α 能实现最小存储开销,而最小化 β(相对于固定值 d)则实现最小修复带宽开销。不难推导出,不可能同时实现 α 和 β 的最小化。这两种极值情况下的编码分别称为最小存储冗余再生码(MSRRC)和最小带宽冗余再生码(MBRRC)。图2给出了当 $B=27000$,$k=10$,$d=18$ 和 $d>18$ 时冗余再生码的存储-带宽平衡曲线[17]。

对于 MSRRC,当最小化 α 后,可得 β 的值:

$$\alpha=\frac{B}{k},\beta=\frac{Bd}{k(d-k+1)} \tag{3}$$

MSRRC 编码中的参数 (α,β,B) 满足式(1),为简化描述编码实现的复杂性,假定 $\beta=1$,则 MSRRC 编码中的 α 和 B 的值为:

$$\alpha=d-k+1,B=k(d-k+1) \tag{4}$$

2 MSRRC 冗余再生码的实现原理

MSRRC 编码利用矩阵实现。再生码矩阵由数据矩阵和修复矩阵生成。数据矩阵是对称矩阵,包含数据本身。修复矩阵预先定义,用于分布数据矩阵中的数据到各存储节点,有助于从节点的子集实现数据重构和失效节点修复。

文件 B 分割并条带化后用向量 $B=[b_1,b_2,\cdots,b_B]$ 表示。利用 $(n\times\alpha)$ 维矩阵 $C(b)$ 表示冗余再生码,$C(b)$ 中的第 i 行 c_i 包含的 α 个元素由节点 i 存储,其用式(5)表示:

$$C(b)=R\cdot D(b) \tag{5}$$

式中,R 是 $(n\times m)$ 维修复矩阵,R 预先定义并独立于数据矩阵 $D(b)$。修复矩阵用于分布包含在数据矩阵中的信息到 n 个节点,这有利于实现数据重构和失效节点修复。$D(b)$ 数据矩阵的维数是 $m\times\alpha$,数据矩阵 $D(b)$ 是由 $b_1,b_2,\cdots,b_B$ 组合而成。数据矩阵 D 具有如下对称性,即 $D^t=D$。把冗余再生码 C 的第 i 行 c_i 称为节点 i 的编码向量,构建的冗余再生码为:

$$C=\{C(c)\mid c\in F_q^B\} \tag{6}$$

再生码的实现分为数据分布、数据重构和失效节点数据修复,原理可参考文献[19]。

3 MSRRC 编码的具体实现

本节根据修复矩阵 R 和数据矩阵 D 生成一个 $[n,k,d]$ 的最小存储冗余再生码 C,为简化实现,假定 $\beta=1$,这里 $\{n,k,d\}$ 满足 $k\leqslant d\leqslant n-1$,其中参数 $\{\alpha,B\}$ 满足等式(3)。为使问题简化,先考虑 $d=2k-2$ 的情况,容易推广到 $d>2k-2$。由等式(3),当 $d=2k-2$ 时,可得 $\alpha=d-k+1=k-1$,从而 $d=2\alpha$,$B=k\alpha=\alpha(\alpha+1)$。因此,$[n,k,d]$MSRRC 编码 C 的参数值为 $(\alpha=k-1,\beta=1,B=\alpha(\alpha+1))$。

3.1 构建数据矩阵

数据矩阵 D 是 $(d\times\alpha)$ 维矩阵,由 $\{b_i\}_{i=1}^{B}$ 中的数据元素组成,并把 $\{b_i\}_{i=1}^{B}$ 中的 B 个元素分成两个子集,每个子集包括 $\binom{\alpha+1}{2}$ 个元素。两个子集的元素分别构成两个 $(\alpha\times\alpha)$ 对称矩阵 $X_1(b)$ 和 $X_2(b)$,从而组成数据矩阵 D,即

$$D(b)=\begin{bmatrix}X_1(b)\\X_2(b)\end{bmatrix}$$

以 $[6,3,4]$MSRRC 编码为例,由等式(4)可知 $B=6$。文件分割成 $B=[b_1,b_2,\cdots,b_B]$,即 $B=[b_1,b_2,\cdots,b_9]$,数据矩阵 $D(b)$ 中的矩阵 X_1 和 X_2 由 $b_1,b_2,\cdots,b_9$ 组合而成,其中

$$X_1=\begin{bmatrix}b_1&b_2\\b_2&b_3\end{bmatrix},X_2=\begin{bmatrix}b_4&b_5\\b_5&b_6\end{bmatrix},$$

显然,X_1 和 X_2 都是对称矩阵,$X_1^t=X_1$,$X_2^t=X_2$。数据矩阵 $D(b)$ 则为:

$$D(b)=\begin{bmatrix}b_1&b_2\\b_2&b_3\\b_4&b_5\\b_5&b_6\end{bmatrix}$$

3.2 构建修复矩阵

修复矩阵 R 是$(n\times d)$维矩阵，并预先定义好。R 由矩阵 W 和 Z 组成，即 $R=[W,ZW]$，其中 W 是$(n\times\alpha)$维矩阵，Z 是$(n\times n)$维对角矩阵，且矩阵 R 的任意 d 行和 W 的任意 α 行线性无关，且 Z 的 n 个对角元素是不同的。

以上节的[6,3,4]MSRRC 编码为例，为满足需求，其修复矩阵采用基于有限域 F_{13} 的(6,4)范德蒙矩阵，令 $\alpha_i(i=1,2,\cdots,6)$分别等于$(\alpha_1,\alpha_2,\cdots,\alpha_6)$，则：

$$W=\begin{bmatrix}1 & \alpha_1\\ 1 & \alpha_2\\ 1 & \alpha_3\\ 1 & \alpha_4\\ 1 & \alpha_5\\ 1 & \alpha_6\end{bmatrix}=\begin{bmatrix}1 & 1\\ 1 & 2\\ 1 & 3\\ 1 & 4\\ 1 & 5\\ 1 & 6\end{bmatrix}$$

$$Z=\begin{bmatrix}\alpha_1^2 & & & & & \\ & \alpha_2^2 & & & & \\ & & \alpha_3^2 & & & \\ & & & \alpha_4^2 & & \\ & & & & \alpha_5^2 & \\ & & & & & \alpha_6^2\end{bmatrix}=\begin{bmatrix}1 & & & & & \\ & 4 & & & & \\ & & 9 & & & \\ & & & 3 & & \\ & & & & 12 & \\ & & & & & 10\end{bmatrix}$$

修复矩阵 R 为：

$$R=[W,ZW]=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 2 & 4 & 8\\ 1 & 3 & 9 & 1\\ 1 & 4 & 3 & 12\\ 1 & 5 & 12 & 8\\ 1 & 6 & 10 & 8\end{bmatrix}$$

3.3 失效节点数据修复

由数据矩阵和修复矩阵可生成最小带宽冗余再生码 C，即 $C=R\cdot D$，C 是(n,d,k)的 MSRRC 矩阵，具有数据重构和失效节点修复特性。同样以[6,3,4]MSRRC 编码进行失效节点数据修复。

$$C=R\cdot D=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 2 & 4 & 8\\ 1 & 3 & 9 & 1\\ 1 & 4 & 3 & 12\\ 1 & 5 & 12 & 8\\ 1 & 6 & 10 & 8\end{bmatrix}\cdot\begin{bmatrix}b_1 & b_2\\ b_2 & b_3\\ b_4 & b_5\\ b_5 & b_6\end{bmatrix}$$

$$=\begin{bmatrix}b_1+b_2+b_4+b_5 & b_2+b_3+b_5+b_6\\ \hline b_1+2b_2+4b_4+8b_5 & b_2+2b_3+4b_5+8b_6\\ \hline b_1+3b_2+9b_4+b_5 & b_2+3b_3+9b_5+b_6\\ \hline b_1+4b_2+3b_4+12b_5 & b_2+4b_3+3b_5+12b_6\\ \hline b_1+5b_2+12b_4+8b_5 & b_2+5b_3+12b_5+8b_6\\ \hline b_1+6b_2+10b_4+8b_5 & b_2+6b_3+10b_5+8b_6\end{bmatrix}$$

C 矩阵各行分别存储到不同节点，即第 i 行 c_i^t 就存储到节点 i，$i\in(1,\cdots,6)$。例如，节点 2 中存储$[b_1+2b_2+4b_4+8b_5\quad b_2+2b_3+4b_5+8b_6]$。现在假若节点 2 失效，通过剩余存活节点重新再生节点 2 的存储数据，其实现过程如下。

置换节点连接 5 个存活节点的任意 4 个(如节点 1,3,5,6)，可得矩阵：

$$M=\begin{bmatrix}b_1+b_2+b_4+b_5 & b_2+b_3+b_5+b_6\\ \hline b_1+3b_2+9b_4+b_5 & b_2+3b_3+9b_5+b_6\\ \hline b_1+5b_2+12b_4+8b_5 & b_2+5b_3+12b_5+8b_6\\ \hline b_1+6b_2+10b_4+8b_5 & b_2+6b_3+10b_5+8b_6\end{bmatrix}$$

修复矩阵 R 预先已定义，因此，置换节点可得 $R_{replace}$($R_{replace}$是R 子矩阵，对应于 R 的第 2、4、5、6 行数据)。$R_{replace}=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & 3 & 9 & 1\\ 1 & 5 & 12 & 8\\ 1 & 6 & 10 & 8\end{bmatrix}$，失效节点 2 的修复向量 $r_2=[1\ 2\ 4\ 8]^t$，对应的 $w_2=[1\ 2]^t$，$z_2w_2=[4\ 8]^t$。

$R_{replace}$ 的逆矩阵：

$$R_{replace}^{-1}=\begin{bmatrix}7/6 & -1/42 & 5/21 & -8/21\\ -1/6 & 1/6 & -2/3 & 2/3\\ -1/12 & 1/12 & 1/6 & -1/6\\ 1/12 & -19/84 & 11/42 & -5/42\end{bmatrix}$$

容易计算出：

$$R_{replace}^{-1}\cdot M\cdot w_2=\begin{bmatrix}b_1+2b_2\\ b_2+2b_3\\ b_3+2b_4\\ b_4+2b_5\end{bmatrix}=\begin{bmatrix}X_1w_2\\ X_2w_2\end{bmatrix}$$

X_1 和 X_2 为可逆矩阵，失效节点 2 的存储数据可表示为：

$$[X_1w_2]^t+z_2[X_2w_2]^t$$
$$=w_2^tX_1+z_2w_2^tX_2$$
$$=[1\quad 2]\cdot\begin{bmatrix}b_1 & b_2\\ b_2 & b_3\end{bmatrix}+4\cdot[1\quad 2]\cdot\begin{bmatrix}b_4 & b_5\\ b_5 & b_6\end{bmatrix}$$
$$=[b_1+2b_2+4b_2+8b_2\quad b_2+2b_3+4b_5+8b_6]$$

从而实现失效节点 2 的数据修复。

4 理论分析

定理 1 最小存储冗余再生码 MSRRC 失效节点数据修复：根据 MSRRC 编码矩阵 C，从$(n-1)$个存活节点中连接任意 $d=2k-2$ 个节点，就能恢复任意某个失效节点。

证明：假定失效节点数据是编码矩阵 C 对应的行$[w_f\ z_fw_f]$，那么存储在失效节点 f 的数据为：$[w_f\ z_fw_f]\cdot D=[w_f^t\quad z_fw_f^t]\cdot\begin{bmatrix}X_1\\ X_2\end{bmatrix}=w_f^t\cdot X_1+z_fw_f^t\cdot X_2$。再生过程中置换节点(用于置换失效节点)需连接 d 个存活节点$\{l_i\mid i=1,2,\cdots,d\}$。置换节点先从 d 存活节点中分别得到 $r_{l_i}^tD(b)$，$\{l_i\mid i=1,2,\cdots,d\}$，聚集各节点数据可计算出 $R_{replace}D(b)\cdot w_f$，其中

$$R_{replace}=\begin{bmatrix}r_{l_1}^t\\ r_{l_2}^t\\ \vdots\\ r_{l_d}^t\end{bmatrix}$$

w_f 是预先定义的修复矩阵 R 中对应于失效节点 f 的向量。因 R 是范德蒙矩阵，$R_{replace}$ 是 R 的子矩阵，显然 $R_{replace}$ 是可逆矩阵。解码过程只需在 $R_{replace}D(b)\cdot w_f$ 的左边乘以 $R_{replace}^{-1}$，从而得到 $D(b)\cdot w_f$，因为 $D(b)=\begin{bmatrix}X_1\\ X_2\end{bmatrix}$，故 $D(b)\cdot$

$w_f=\begin{bmatrix}X_1\cdot w_f\\X_2\cdot w_f\end{bmatrix}$，又因为 X_1 和 X_2 是对称矩阵，因此，置换节点能得到$(X_1\cdot w_f)^t+z_2(X_2\cdot w_f)^t=w_f^tX_1+z_2w_f^tX_2$，即为失效节点数据。

定理 2 最小存储冗余再生码 MSRRC 的数据重构：根据 MSRRC 编码 C，连接任意 k 个节点就能恢复整个数据文件 B，即对 C 的 k 行进行线性运算，就能恢复文件 B。

证明：汇聚节点 $R_{DC}=[W_{DC}\quad Z_{DC}W_{DC}]$，其中 R_{DC} 是修复矩阵 R 的 $k\times d$ 维子矩阵，矩阵 R_{DC} 的 k 行对应于与汇聚节点连接的 k 个节点，汇聚点 DC 可得到如下等式：

$$R_{DC}D(b)=[W_{DC}\quad Z_{DC}W_{DC}]\begin{bmatrix}X_1\\X_2\end{bmatrix}$$
$$=[W_{DC}X_1+Z_{DC}W_{DC}X_2]$$

等式右边乘以 W_{DC}^T 可得：

$$[W_{DC}X_1+Z_{DC}W_{DC}X_2]W_{DC}^T$$
$$=W_{DC}X_1W_{DC}^T+Z_{DC}W_{DC}X_2W_{DC}^T$$

定义矩阵 U 和 V 为：

$U=W_{DC}X_1W_{DC}^T$，$V=W_{DC}X_2W_{DC}^T$

式中，矩阵 U 和 V、X_1 和 X_2 都是对称矩阵。由矩阵 U 和 V，汇聚点 DC 能获得矩阵 $U+Z_{DC}V$，其中矩阵的第(i,j)个元素为：

$U_{ij}+z_iV_{ij}$，$1\leqslant i,j\leqslant k$

因为 U 和 V 是对称矩阵，故矩阵的第(j,i)个元素为：

$U_{ji}+z_jV_{ji}=U_{ij}+z_jV_{ij}$

对角矩阵 Z 的元素 z_i 是不同的，由上面两等式，汇聚节点 DC 能得 U_{ij} 和 V_{ij} 的值$(i\neq j)$。首先考虑矩阵 U，让 $W_{DC}=\begin{bmatrix}w_{l_1}^t\\w_{l_2}^t\\\vdots\\w_{l_\alpha+1}^t\end{bmatrix}$，已知矩阵 U 的所有非对角元素，因此，矩阵 U 的第 l_i 行元素为 $w_{l_i}^tX_1[w_{l_1}\quad\cdots\quad w_{l_{i-1}}\quad w_{l_{i+1}}\quad\cdots\quad w_{l_{\alpha+1}}]$，此矩阵的右边是满秩，汇聚点 DC 能得到：

$w_{l_i}^tX_1$，$1\leqslant l_i\leqslant\alpha+1$

当选择前 α 个元素，DC 能得到：

$$\begin{bmatrix}w_{l_1}^t\\w_{l_2}^t\\\vdots\\w_{l_\alpha+1}^t\end{bmatrix}\cdot X_1$$

此矩阵的左边也是满秩的，因此 DC 能恢复 X_1。同理，利用 V 的非对角元素，DC 也能恢复 X_2，从而重构数据 $D(b)=\begin{bmatrix}X_1\\X_2\end{bmatrix}$，得证。

5 实验分析

通过在实际的分布式系统进行模拟实验，追踪节点的可用性来获取不同的参数值，收集不同修复策略的平均带宽以及文件的平均可用性，从而比较不同的策略的性能。虽然评估方法对实际的存储系统进行了简化，但能直接比较不同的修复策略。

5.1 实验模型

实验使用模型获得系统的平均带宽以及文件的平均可用性。模型利用两个关键性的参数：F 和 A，F 表示单位时间存储节点数据永久失效率(死亡概率)，A 表示单位时间数据节点的可用率。$1-A$ 则表示节点暂时失效率(暂时离线率)。这里假定某节点是否可用独立于其它节点的可用性。在这些假定下，计算维护大小为 B 个字节文件的不同冗余策略的可用性和维护带宽(注意除了 MSRRC 编码外，修复网络带宽等于需要置换的冗余数量，即存储文件大小的 F 倍)。

复制修复：如果存储文件的 S 个副本，则总的存储大小为 $S\cdot B$，每单位时间必须置换 $F\cdot S\cdot B$。当副本丢失时，文件不可用的发生概率为$(1-A)^S$。

混合修复：如果存储文件的完整副本以及该文件的(n,k)纠删码，其中 $n=k\cdot(S-1)$，则总计需存储 $S\cdot B$ 个字节，每单位时间需传递 $F\cdot S\cdot B$ 个字节。只有当文件副本和 k 个以上纠删码分片都不可用时，文件才不可用，其不可用的概率是：$(1-A)\cdot\sum_{i=0}^{k-1}\binom{n}{i}A^i(i-A)^{n-i}$。

最小存储冗余再生码 MSRRC：(n,k)MSRRC 编码其冗余度为 $S=n/k$，共需存储 $S\cdot B=n\cdot B/k$，因此，每单位时间需置换 $F\cdot S\cdot B$ 字节。如果把 λ_{MSRC} 作为需要传递的信息量与分块大小 B/k 的比值，可求得：

$$\lambda_{MSRC}=\frac{(n-1)\beta_{MSRC}}{\frac{B}{k}}=\frac{(n-1)\frac{2B(n-1)}{k(n-k)}}{\frac{B}{k}}=\frac{(n-1)}{(n-k)}$$

因此，当新加入节点连接到 $d=n-1$ 个节点时，替换一个分片最少需要传递 λ_{MSRC} 倍分片大小的数据量。单位时间需发送 $F\cdot S\cdot B\cdot\delta_{MSR}$ 字节，其不可用性为：

$$U=\sum_{i=0}^{k-1}\binom{n}{i}A^i(i-A)^{n-i}。$$

5.2 实验分析

实验方法与文献[19]类似，即追踪节点的可用性获取不同的参数值，收集不同修复策略的平均带宽以及文件的平均可用性，从而比较不同策略的性能。虽然评估方法对实际的存储系统进行了简化，但能使我们直接比较不同的修复策略，实验的主要参数如表 1 所列，测试结果如图 3 和图 4 所示。

表 1 MSRRC 编码实验的主要参数

参数	值
节点数目	>300
每个节点容量|GB	>50
节点永久失效率 F	0.02
平均节点可用率	0.98
文件大小|GB	1
网络带宽|(MB·s^{-1})	>100
互 Ping 追踪的间隔时间|Minutes	15 分钟
测试时间|T	10Day

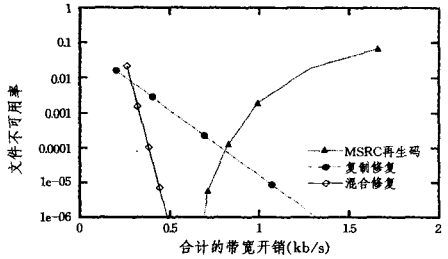

图 3 $n=10,k=5$ 时不同冗余策略的带宽/可用性曲线

(下转第 207 页)

lications Ltd,2000

[5] 罗文伯. 社会网络视角下的微博研究[J]. 今传媒,2013(2):108-109

[6] 刘军. 整体网分析讲义——UCINET 软件应用[C]//第二届社会网络与关系管理研讨会. 哈尔滨:哈尔滨工程大学社会学系,2007:119

[7] 罗家德. 社会网络分析讲义[M]. 北京:社会科学文献出版社,2005:150-151

[8] 丁兆云,贾焰,周斌,等. 社交网络影响力研究综述[J]. 计算机科学,2014,41(1):48-53

[9] 党洪莉,孙红霞. 图书情报学博客的社会网络分析[J]. 情报杂志,2009(1):180-181

[10] 袁圆,孙霄凌,朱庆华. 微博用户关注兴趣的社会网络分析[J]. 现代图书情报技术,2012(2):68-75

[11] 魏顺平. 社会网络分析及其应用案例[J]. 现代教育技术,2010,20(3):29-34

(上接第 194 页)

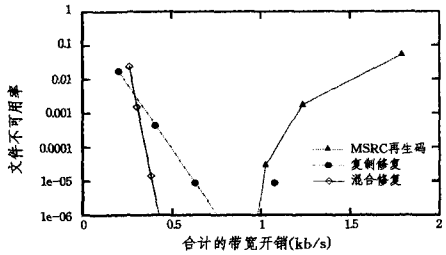

图 4 $n=20, k=10$ 时不同冗余策略的带宽/可用性曲线

从图中可以得出,要获得同样的数据可用性,MSRRC 编码比其它策略需要更多的网络带宽,比如在 0.0001 位置(即若需要 99.9999%的数据可用性),但存储开销则大大减少。但值得注意的是:MSRRC 冗余再生码只需维护一种冗余类型,复合策略需同时维护完整副本和纠删码分块,故冗余再生码比复合策略的系统设计简单,又由于冗余再生码具有纠删码的一切属性,因此其整体可用性是复制策略所无法比拟的。

结束语 冗余再生码(Regenerating Codes)是一种分布式存储编码,能优化失效节点修复中的带宽与存储开销。论文根据冗余再生码数据再生时的极值点:最小存储再生点(MSR),通过矩阵运算实现最小存储冗余再生码 MSRRC。文中详细给出再生码的数据重构和失效节点再生的实现原理,并利用实例描述了冗余再生码的实现过程,在理论上证明了实现原理的可靠性,文中还给出了实验运行结果。

参 考 文 献

[1] Patterson D A, Gibson G, Katz R H. A case for redundant arrays of inexpensive disks (RAID) [C] // Proc. ACM SIGMOD Chicage. USA, June 1988:109-116

[2] Kubiatowicz J, Bindel D, Chen Y, et al. OceanStore: An Architecture for Global-Scale Persistent Store[C] // Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000). 2000:190-201

[3] Bhagwan R, Kati K, Cheng Y-C, et al. Total recall: System support for automated availability management [C] // Proc. of ACM/USENIX NSDI'04. 2004:337-350

[4] Wu Y, Dimakis A G, Ramchandran K. Deterministic regenerating codes for distributed storage[C] // Allerton Conference on Control, Computing and Communication. Monticello, October 2007

[5] Wang Z, Dimakis A G, Bruck J. Rebuilding for Array Codes in Distributed Storage Systems[C] // Workshop on the Application of Communication Theory to Emerging Memory Technologies (ACTEMT). Dec. 2010

[6] Akhlaghi S, Kiani A, Ghanavati M R. A Fundamental Tradeoff between the download cost and repair bandwidth in distributed Storage systems[C] // Proceeding of IEEE International Symposium on network Coding (NetCod). Toronto, Jun. 2010

[7] Shah N B, Rashmi K V, Kumar P V. A Flexible Class of generating Codes for Distributed Storage[C] // Proceeding of IEEE International symposium on information theory (ISIT). Austin, Jun. 2010:1943-1947

[8] Gaston B, Pujol J. Double Circulate Minimum Storage generating Codes. 2010[OL]. http://arXiv:1007.2401[cs. IT]

[9] Wang Z, Mateescu R, Dimakis A G, et al. Array codes for distributed storage: Results and open problems [J]. Information Tehory and Application. ITA, 2010

[10] Wu Y. Existence and construction of capacity-achieving network codes for distributed storage[J]. Journal on Selected Areas in Communications, 2010, 28(2):277-288

[11] Li J, Yang S, Wang Xin, et al. Tree structured Data Regeneration in Distributed Storage Systems with Regenerating Codes [C] // Proceedings of IEEE INFOCOM. 2010

[12] Shah N B, Rashmi K V, Kumar P V, et al. Distributed Storage Codes with Repair-by-Transfer and Non-achievability of Interior Points on the Storage-Bandwidth Tradeoff [C] // Allerton Conf. . Urbana-Champaign, Sep. 2009

[13] Dimakis A G, Godfrey P G, Wainwright M J, et al. Network coding for peer-to-peer storage[C] // Proceeding of INFOCOM. Anchorage, Alaska, May 2007

[14] Wu Y, Dimakis A G, Ramchandran K. Deterministic regenerating codes for distributed storage[C] // Allerton Conference on Control, Computing, and Communication. Monticello, October 2007

[15] Shah N B, Rashmi K V, Kumar P V, et al. Distributed storage codes with repair-by-transfer and non-achievability of interior points on the storage-bandwidth tradeoff[J/OL]. http://arxiv.org/abs/1011.236

[16] Wu Y, Dimakis A G, Ramchandran K. Deterministic Regenerating codes for distributed storage[C] // Proc. Allerton Conf. . Urbana-Champaign, Sep. 2007

[17] Ramabhadran S, Pasquale J. Analysis of durability in replicated distrusted storage systems[C] // 2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS). Atlanta, GA, April 2010:1-12

[18] 王禹,赵跃龙,侯昉. 基于副本管理的 P2P 存储系统的可靠性问题研究[J]. 华南理工大学学报,2011,39(2):148-152

[19] 王禹,赵跃龙,侯昉,分布式存储系统最小带宽再生码研究[J]. 小型微型计算机系统,2012,33(8):1710-1714