View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Xiamen University Institutional Repository

一种数据网格容灾存储模型及其数据失效模型

曲明成[1], 吴翔虎[1], 廖明宏[2], 张　银[1], 杨孝宗[1], 左德承[1]
(1. 哈尔滨工业大学计算机科学与技术学院, 黑龙江哈尔滨 150001; 2. 厦门大学软件学院, 福建厦门 361005)

摘　要: 可靠性较高的数据网格多采用双副本容灾可以保证节点在灾难发生时进行有效恢复, 但由于节点数据存储量较大, 当节点发生灾难时, 从一个节点进行数据恢复速度较慢, 导致在数据恢复期备份节点发生失效的概率较大. 针对这些问题, 本文给出一个容灾存储模型, 基于该模型推导出一个数据失效模型, 理论证明该失效模型的数据失效概率明显小于双副本容灾方式的数据失效概率, 同时在灾难发生时又能达到较快的恢复速度. 设计了仿真实验, 将模型的数据失效概率与双副本失效概率进行了比较, 实验结果与理论推导一致, 证明了存储模型和数据失效模型的正确性和有效性. 最后给出进一步研究思路.

关键词: 数据网格容灾; 存储模型; 数据失效模型; 并行数据传输; GridFTP
中图分类号: TP311　　文献标识码: A　　文章编号: 0372-2112 (2010) 02-0315-06

A Disaster-Tolerant Storage Model and a Low Data Failure Model for Data Grid

QU Ming-cheng[1], WU Xiang-hu[1], LIAO Ming-hong[2], ZHANG Yin[1], YANG Xiao-zong[1], ZUO De-cheng[1]
(1. *School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China;*
2. *Software School, Xiamen University, Xiamen, Fujian 361005, China*)

Abstract: With double copies(DC) for Disaster-Toelerance, the data grid with higher reliability can ensure the effective recovery, reduce the data failure probability(DFP) when the disaster occurs, and can provide greater network bandwidth. But because of the large data storage of the nodes, when the disaster occurs to the nodes, data recovery from one node is a little slow, which leads to the higher failure probability of the backup nodes during the data recovery period, therefore, the data failure probability gets higher. To solve these problems, a Disaster-Tolerant Storage Model(DTSM) is put forward, based on this model, a Data Failure Model(DFM) is built. Theoretical proof shows that the DFP of DFM is lower than that of DC and the recovery speed is faster when the disaster happens. To make a comparison between the data failure probabilities of two models, a simulating experiment is designed. The experimental result is consistent with the theoretical deduction, which proves that DTSM and DFM are correct and effective. Finally a further research idea is proposed.

Key words: disaster tolerant for data gird; storage model; data failure model; parallel data tramsmission; gridFTP

1　引言

数据网格的概念来自网格(Grid), 它是网格技术在数据管理方面的应用和实现, 即是为了建立网格环境下透明访问异构数据资源的新的体系结构[1]. 确保数据网格中存储的海量数据的完整性、可用性是数据网格研究的关键问题. 由于网格中节点的不可靠性, 在节点发生容灾时, 必须确保数据的完整可用, 因此必须尽快地选出新节点并进行灾难节点的数据备份恢复[2].

编码解码技术的研究主要仿效于通信领域, 而将其应用于分布式、动态的网格容灾其本身解码速度较慢并且实现复杂, 实际应用中可行性较差[3]. 因此目前针对节点整体可靠性较高的网格, 容灾都是通过在异地建立和维护冗余数据, 利用地理分散性来保证数据对灾难事件的抵御能力. 根据容灾的概念可知, 网格容灾的核心就是增加数据冗余度, 当灾难发生时, 让数据的副本被同时毁坏的概率降到可以接受的程度, 降低数据副本被同时毁坏的概率, 提升数据的恢复速度, 保证整体数据具有较高的可用性、较低的失效概率[4].

现有的针对可靠性较高的网格, 其容灾技术实现方法虽然各不相同, 但主要是基于多副本, 较多的为双副本[2,4,5]. 使用双副本容灾, 当发生灾难时(节点损坏或

收稿日期: 2008-11-19; 修回日期: 2009-06-28
基金项目: 国家自然科学基金(No. 60533110)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

短期内不可用), 从一个副本节点进行数据恢复, 由于受到传输链路和节点网络带宽限制, 其恢复速度明显较慢, 由于数据恢复期较长, 在恢复期内备份节点发生失效的概率随之增大. 如何在节点发生灾难后对数据进行快速恢复以降低整体数据失效概率是数据网格亟待解决的关键问题.

GridFTP 是为网格快速传输而设计的传输协议, 针对 GridFTP 协议, 很多学者分别基于 Linux、Unix 和 Windows 操作系统进行实现方法研究, 取得了良好的应用效果[6~8]. GridFTP 提供了条状数据传输方式, 即 GridFTP 客户端可以并行的从多个 GridFTP 服务器端下载不同数据块. 基于 GridFTP 协议出现了很多的并行传输算法, 这大幅度提高了网格中数据的传输速度[9~11].

基于上述理论和存在的问题, 本文提出了一个数据网格的容灾存储模型, 基于存储模型推导出一个数据失效模型. 存储模型允许任意 P 个节点的同时失效, 失效模型使数据的整体失效概率明显小于双副本容灾模式, 同时存储模型能够达到较快的数据恢复速度. 与双副本容灾模式进行了数据失效概率和恢复速度比较, 取得了较好的效果. 并给出模型 $p=2$ 时的实验思路, 以及后续研究内容.

2 容灾存储模型

定义 1(ω等分), 令总数据量为 M, 对整个数据进行等分, 令分割的份数等于 $k(k-1)\omega$, k 为副本节点的个数. 分割方式为: 先将数据等分成 $k(k-1)$ 份, 再将每一份等分成 ω份, 这里 ω是一个可变参数. 则每一份的数据量 m 等于: 见式(1).

$$m=\frac{M}{k(k-1)\omega} \tag{1}$$

定义 2(本地数据), 将定义 1 中分割的 $k(k-1)\omega$ 份数据平均分配到 k 个节点上, 则每个节点存储 $(k-1)\omega$ 份数据. 称这些数据为节点 N_i 的本地数据 L_i. 注: L 表示本地数据, L_i 表示节点 i 的本地数据.

定义 3(本地数据虚拟组), 将节点 N_i 存储的本地数据块进行虚拟组划分, 即将 ω个数据划为一组, 将划分后的组进行节点内编号. 由定义 2 知, 一个节点本地数据可以划分的虚拟组数为 $(k-1)$ 个, 令虚拟组为 G_i^j $(0\leqslant i\leqslant k-1, 0\leqslant j\leqslant k-2)$. 令 G_i 表示节点 N_i 的所有虚拟组, G 表示当前虚拟组.

定义 4(剩余节点集合), 令刨除节点 N_i 后的所有参与存储的节点集合为剩余节点集合, 即

$$\bigcup_{y=0}^{k-2}\overline{N}_i^y=\bigcup_{j=0,j\neq i}^{k-1}N_j,\ \overline{N}_i^y=N_j,\begin{cases}y=j, & \text{if } i<i\\ y=j-1, & \text{if } j>i\end{cases}$$

定义 5 (交叉存储), 将节点 N_i 的数据 G_i 存储到刨除 N_i 的其他 $k-1$ 个节点 $\overline{N}_i^e$ 上, 即 $G_i\rightarrow\overline{N}_i^e$, 并满足如下规则, $(\bigcup_{r=e}^{w}(G_i^r\rightarrow\overline{N}_i^e))\mid_{e=0}^{k-2}\{w=(e+(p-1))\bmod k, p\leqslant k-1\}$, p 为指定的常数, "$\rightarrow$" 表示存储到, 称这种存储方式为交叉存储.

容灾存储模型 节点 N_i 存储的所有数据 A_i 包括本地数据 L_i, 和其他节点的交叉存储数据 O_i, 有

$$A_i=(L_i)\cup(O_i)=\left(\bigcup_{j=0}^{k-2}G_i^j\right)\cup\left(\bigcup_{(a=0,a\neq ij)}^{k-1}\left(\bigcup_{r=e}^{w}G_a^r\right)\right)\begin{cases}e=i-1, & a<i\\ e=i, & a>i\end{cases}$$

$$\text{and}\begin{cases}w=(e+(p-1))\bmod k\\ p\leqslant k-1,\end{cases}$$

p 为指定的常数, 称其为容灾存储模型. 注: O 表示本地数据, O_i 表示节点 i 的本地数据.

定理 1 (P 完整性), 如数据满足存储模型, 则当有任意 p 个节点不可用时, 其余 $k-p$ 个节点中数据的并集仍然等于整体数据量 M, 即完整的数据, 称这种性质为 P 完整性.

证明 对任意的一个局部数据 G_i^x 进行讨论, 其中 $i\in(0,k-1)$, $x\in(0,k-2)$, G_i^x 表示任意一个节点 N_i 的任意一个本地虚拟组.

由定义 5 可知, $e\in(0,k-2)$, $r\in(e,w)$, 又 $\{w=(e+(p-1))\bmod k, p\leqslant k-1\}$, 则 e 到 w 共有 p 个数. 取出一个子集 $e\in(x-(p-1),x)\subset(0,k-2)$, 因为对于 e 的每次取值, r 需要从 e 开始取 p 个值. 当 e 由 $x-(p-1)$ 增加到 x 过程中, r 的取值范围分别为: $\{x-(p-1),x\}$, …, $\{x,x+(p-1)\}$, x 重复了 p 次, 又由于 e 经历了 P 个值的变化, 即 G_i^x 有 p 次存储到节点 $\overline{N}_i^e$ 上, 这里 $e\in(x-(p-1),x)$. 加上 G_i^x 虚拟组所在的节点, 共有 $p+1$ 个节点存储 G_i^x. 因此如果有任意的 p 个节点不可用, 仍然有一个存储 G_i^x 的节点. 由于 G_i^x 具有任意性, 则所有节点的所有虚拟组均满足上述推导. 证毕.

引理 1 满足数据存储模型的数据, 其整体存储数据量可以表示为: $k(1+p)(k-1)\omega$.

证明 由存储模型定义可以推出:

$$\sum_{i=0}^{k-1}A_i=\sum_{i=0}^{k-1}\left((k-1)\omega\mid_{L_i}+(1+p)\omega(k-1)\mid_{O_i}\right)$$

$$=k(k-1)(1+p)\omega \qquad \text{证毕.}$$

定义 6 (存储空间使用量率, S), 采用存储模型总体数据存储量与采用 k 个完整副本存储的总数据量比值定义为存储空间使用量率 S. 由定义 1 和引理 1 可得 S 如式(2).

$$S=\frac{k(k-1)(1+p)\omega}{k^2(k-1)\omega}=\frac{1+p}{k} \tag{2}$$

算例 1 给定 $k=4$, $P=2$, $\omega=4$. 则满足容灾存储模型的数据分配推演过程如下:

① 由定义 1 得: 数据分割份数 $=4*(4-1)*4=$

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

48. 令这些数据块分别为: (1), (2), …, (48) .

②由定义 2 得: L_0= {(1), (2), …, (12)}; L_1= {(13), (14), …, (24)}; L_2= {(25), (26), …, (36)}; L_3= {(37), (38), …, (48)}.

③由定义 3 得: (a) G_0^0= {(1), (2), (3), (4)}, G_0^1= {(5), (6), (7), (8)}, G_0^2= {(9), (10), (11), (12)}; (b) G_1^0= {(13), (14), (15), (16)}, G_1^1= {(17), (18), (19), (20)}, G_1^2= {(21), (22), (23), (24)}; (c)、(d) 略

④由定义 4 得: (a) $\overline{N}_0^0$= N_1, $\overline{N}_0^1$= N_2, $\overline{N}_0^2$= N_3; (b) $\overline{N}_1^0$= N_0, $\overline{N}_1^1$= N_2, $\overline{N}_1^2$= N_3; (c)、(d) 略

⑤由定义 5 得: (a) $G_0 \rightarrow \overline{N}_0^0$= $G_0 \rightarrow N_1$= {G_0^0, G_0^1} $\rightarrow N_1$, $G_0 \rightarrow \overline{N}_0^1$= $G_0 \rightarrow N_2$= {G_0^1, G_0^2} $\rightarrow N_2$, $G_0 \rightarrow \overline{N}_0^2$= $G_0 \rightarrow N_3$= {G_0^2, G_0^0} $\rightarrow N_3$; (b) $G_1 \rightarrow \overline{N}_1^0$= $G_1 \rightarrow N_0$= {G_1^0, G_1^1} $\rightarrow N_0$, $G_1 \rightarrow \overline{N}_1^1$= $G_1 \rightarrow N_2$= {G_1^1, G_1^2} $\rightarrow N_2$, $G_1 \rightarrow \overline{N}_1^2$= $G_0 \rightarrow N_3$= {G_1^2, G_1^0} $\rightarrow N_3$; (c)(d) 略

⑥由容灾存储模型得出如下的一个数据分配:

A_0= {(1), (2), …, (12); (13), (14), (15), (16); (17), (18), (19), (20); (25), (26), (27), (28); (29), (30), (31), (32); (37), (38), (39), (40); (41), (42), (43), (44)};

A_1= {(13), (14), …, (24); (1), (2), (3), (4); (5), (6), (7), (8); (29), (30), (31), (32); (33), (34), (35), (36); (41), (42), (43), (44); (45), (46), (47), (48)};

A_2, A_3 略.

3　数据失效模型

3.1　失效模型(P= 1)

节点失效概率为节点发生失效的概率, 当节点出现失效时该节点数据将不再可用(或短期不再可用), 数据必须恢复到新节点. 数据失效概率为数据非完整的概率, 即当从现有节点中无法取到完整数据时, 则发生数据失效. 为简化模型的推导, 令各节点间传输速度相同, 比如从一个节点传输数据速度为 V, 则如果从两个节点同时传输就为 $2V$, 即节点的接入带宽远大于网络链路的传输带宽, 从而并行数据传输的速度不会超过节点接入带宽.

p= 1 时, 根据定义 6 可知, 存储的数据量为两倍完全副本. 那么此时采用模型与采用双副本存储数据, 当发生节点灾难时, 恢复模式如图 1 所示. 图 1 左图为 p = 1 时存储模型的恢复方式, 节点 N_i 发生灾难, 需要从其他 $k-1$ 个节点同时恢复其存储的数据到新节点 X, 由定义 2、5 可知, 每个节点存储的数据量为完全副本的 $2/k$; 在右图中为采用双副本存储, 如果节点 B 发生灾难, 则只能从节点 A 恢复数据到新节点 C, 两种方案恢复速度比值为 $(k-1)$. 在那么两种方案中, 对于只有一个节点灾难的恢复时间比值为 $2/(k(k-1))$.

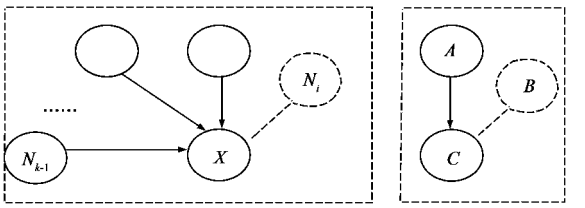


图1　P=1时失效模型与双副本存储数据恢复

令节点的单位时间失效概率为 F. 如果有节点发生灾难, 则自动选取一个新节点, 将备份节点的数据恢复到新节点, 令双副本存储恢复数据的时间与单位时间的比值为 θ(即进行一次完全数据恢复). 令双副本冗余存储的数据失效概率为 α, 模型(p= 1)的失效概率为 β. 取两种方式的临界条件, 即再有一个节点失效则数据失效(不完整), 两种方式都最多允许一个节点失效, 那么在其数据恢复期内, 不允许再发生其他节点失效, 即: 对于模型在 $2\theta/(k(k-1))$ 时间内, 超过一个节点失效, 则整体数据(失效)不完整; 对于双副本, 在 θ 时间内超过一个节点失效, 则整体数据(失效)不完整. 基于容灾存储模型、并行传输理论和泊松分布推出数据失效模型如式(3)、(4).

$$\beta = 1 - \sum_{i=0}^{1} C_k^i f^i (1-f)^{k-i},\ k \geqslant 2$$

$$f = \left(\frac{\theta}{k-1} \cdot \frac{2}{k}\right) F = \frac{2\theta F}{k(k-1)},\ \alpha = (\theta F)^2 \qquad (3)$$

$$\beta = 1 - \underbrace{\left(\left(1 - \frac{2\theta F}{k(k-1)}\right)^k + \frac{2\theta F}{(k-1)}\left(1 - \frac{2\theta F}{k(k-1)}\right)^{k-1}\right)}_{D(k)} = 1 - D(k),\ k \geqslant 2 \qquad (4)$$

定理**2**　当 $k \geqslant 2$ 时, β 为单调减函数, 令常数 e 为自然对数函数的底数.

证明**:** 由泊松分布可知, 式(4)中的 $D(k)$ 部分可以进行如下变换:

$$D(k) = e^{-\lambda}\left(1 + \lambda\right),\ \lambda = fk = \frac{2\theta F}{k-1},\ \lambda' = -\frac{2\theta F}{(k-1)^2}$$

$$D(k)' = -e^{-\lambda}\lambda'(1+\lambda) + e^{-\lambda}\lambda' = -e^{-\lambda}\lambda'\lambda = e^{-\lambda}\frac{4(\theta F)^2}{(k-1)^3} > 0$$

因为 $D(k)$ 的一阶导数大于 0, 所以 $D(k)$ 为单调增函数, 进一步得出 $\beta = 1 - D(k)$ 为单调减函数.　证毕.

定理 **3**　当 $k \geqslant 2$ 时, α/β 为单调增函数.

证明**:** 当 k= 2 时, 式(4)转化为:

$$\beta = 1 - \left(\left(1 - \frac{2\theta F}{2(2-1)}\right)^2 + \frac{2\theta F}{2-1}\left(1 - \frac{2\theta F}{2(2-1)}\right)^{2-1}\right) = 1 - (1 - 2\theta F + (\theta F)^2 + 2\theta F - 2(\theta F)^2)$$

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

$= (\theta F)^2 = \alpha$ (5)

由定理 2 和式(5)可以推出 α/β 单调递增. 证毕.

由定理 3 可知, 当 $k > 2$ 时, α 的失效概率大于 β 的数据失效概率, 同时随着节点的增加, 两者的失效概率比值逐渐增大.

3.2 失效概率($P = 2$)

$p = 2$ 时, 存储量为三倍副本, 即如果采用完整副本备份那么就需要三个副本存储. 此时的数据失效概率很难用一个数学模型进行描述, 其恢复过程示例如图 2 所示. t_0 时节点 3 失效, 迅速从节点 1、2 进行新节点的数据恢复, 到 t_2 时可以恢复完成, 但是在 t_1 时节点 2 又出现失效, 那么节点3恢复时间延长至t_3, 节点2恢复时间至 t_4, 那么在区间内(t_1, t_3), 如果节点 1 再发生失效, 则整体数据失效. 对于采用容灾模型存储, 数据恢复存在同样的情况, 因此其数据失效概率较难用数学模型描述, 但是可以通过大量实验取得.

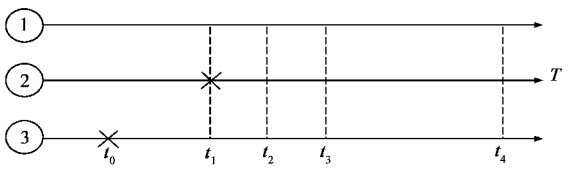


图2 三副本存储数据恢复

4 仿真实验

实验采用单机仿真方法. α 为平方项, 对可靠性较高的网格其 θF 是一个较小的数, 忽略 β 中的三次和更高次项(检测 α/β). 首先使用 Matlab 求出定理 3 的 α/β 理论走势, 然后将其与实验得出结果进行对比分析(图 4); 并对理想的数据恢复时间进行分析(图 5). 实验方法如下:

取单位时间节点失效概率 $f = 0.0005$, 以 2000 个时间单位为一组, 因为 $2000 * 0.0005 = 1$, 所以在 2000 个时间单位内将有一个节点出现失效. 令实验总时间单位数为 20 万, $\theta = 550$, 以 11 个节点的 β 求解过程为例, 实验原理如图 3 所示. 假设失效为不可恢复, 即节点不可用.

11个节点在20万个单位时间内的正常状态用0标识, 每节点在 2000 个时间单位随机的出现一次失效, 用 1 标识. 图中的逗点表示省略. W_i 表示失效窗口, 窗的高度 H 为 11(节点数), 宽度 U 为 $10 = 2\theta/(k(k-1))$, 以步长为 1 向右移动, 每移动一次窗口, 检测窗口内 1 的个数, 如果小于等于 1 个(p 等于 1), 则将失效检测变量 g 加 1(初始时为 0), 表示数据完整而没有失效. 当窗口向右移动到尽头时, 按如下公式计算失效概率: 移动的总步数 $j = 2^* 10^5 - 10$, 出现失效的次数 $C = 2^* 10^5 - 10 - g$, 则 $\beta = C/j$.

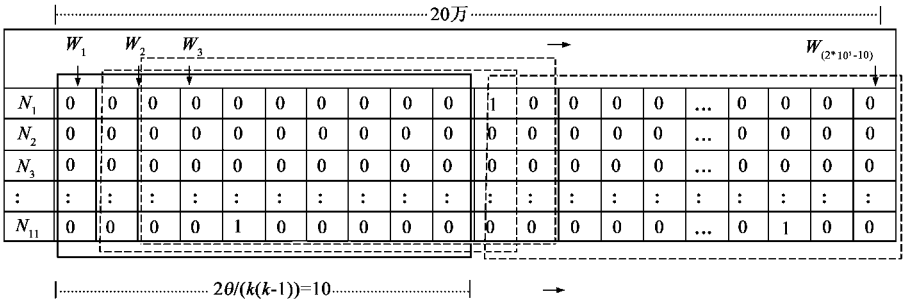
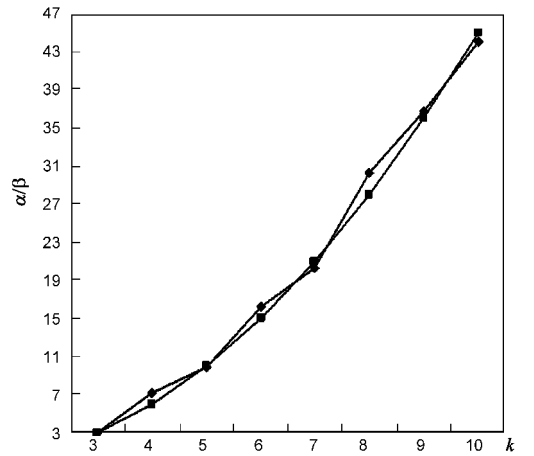


图3 失效窗口求解失效概率

通用的求解方法如公式(6), 其中 U 如果是非整数时, 令 $U = \mathrm{floor}(U)$, floor 为做取底处理. 按照这种方式求出 k 为不同值时数据失效模型的数据失效概率, 并算出实验中双副本数据失效概率与模型的失效概率比值, 同时使用 matlab 求解出定理 3 理论数据失效概率比值, 绘出曲线图 4.

$$\beta = \frac{\left(2^* 10^5 - U\right) - g}{\left(2^* 10^5 - U\right)}, \quad U = \frac{2\theta}{k(k-1)} \tag{6}$$

从图 4 中可以看出实际曲线的走势与模型的理论走势相符, 但是存在微小误差. 模型在没有损失存储空间的情况下, 其失效概率明显小于双副本容灾方案的



实验	2.98	7.12	9.95	16.20	20.28	30.22	36.70	43.98
理论	3.00	6.00	10.00	15.01	21.01	28.01	36.01	45.01

图4 α/β与k的关系

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

失效概率. 根据文献[6~ 12]的 GridFTP 并行数据传输协议和算法, 以节点传输速度相同为前提, 存储模型的数据恢复时间走势如图5 所示. 可以看出模型的数据恢复时间随着节点数的增多较双副本优势尤为显著.

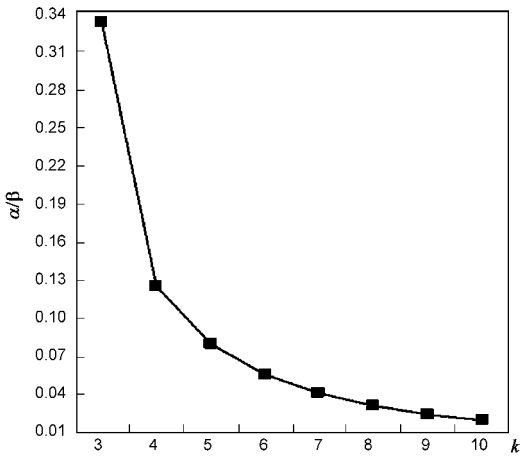


图5　α与β恢复时间比值

当多个数据存储节点的传输速度之和超过恢复节点的接入带宽时, 继续增加存储节点将不能提升数据的恢复速度, 并且会降低数据的失效概率, 因此根据实际情况可以控制存储模型中的 k 值, 将几个节点归为一组形成一个存储联盟, 使在联盟内满足存储模型.

实例如下: 假设一类数据需要 3 个节点存储, 采用双副本容灾则需要 6 个节点. 令数据分别为(1)(2)(3)(4)(5)(6)(7)(8)(9)(10)(11)(12). 双副本和容灾模型(p = 1)存储方式比较如表 1 所示. 三副本与模型的存储方式比较以表 1 和模型定义为参考可以得出, 过程略.

表 1　双副本和容灾模型存储方式比较

节点	N_1	N_2	N_3	N_4	N_5	N_6
双副本	原始			副本		
	(1)(2)	(5)(6)	(9)(10)	(1)(2)	(5)(6)	(9)(10)
	(3)(4)	(7)(8)	(11)(12)	(3)(4)	(7)(8)	(11)(12)
模型(p= 1)	组 1			组 2		
	(1)(2)	(3)(4)	(5)(6)	(7)(8)	(9)(10)	(11)(12)
	(3)(5)	(1)(6)	(4)(2)	(9)(11)	(7)(12)	(10)(8)

5　结论

本文针对数据网格双副本容灾模式的数据恢复速度较慢以及数据失效概率较大等问题, 提出了一个数据网格容灾存储模型和数据失效模型. 通过理论推导和仿真实验证明了存储模型具有较低的数据失效概率、较快的数据恢复速度, 同时较双副本容灾方式没有任何存储空间的浪费. 这表明该容灾存储模型以及数据失效模型是正确的, 并且存储模型具有较强的可行性和实用性.

6　后续研究

针对 p= 2 时, 设计一套合理的失效检测算法, 并进行实验验证, 并将三副本存储与容灾模型存储的失效概率进行比较. 模型中的虚拟组、ω 为后续的副本部署和并行传输调度算法做了准备工作, 即研究多副本部署和并行传输中如何达到速度优化和存储空间节约的双重优化问题.

参考文献:

[1] 陈磊, 李三立. 数据网格中一种填空式副本分配算法[J]. 电子学报, 2006, 34(11): 1951- 1954.
Chen Lei, LI Sanr li. A calking dynamic replication distribution algorithm in data grid[J]. Acta Eelectronica Sinica, 2006, 34(11): 1951- 1954. (in Chinese)

[2] Yu Xiangzhan, Wu Guanjun, et al. An disaster tolerance model based on dataflow replication[A]. Proceedings of the 2008 IEEE International Conference on Information Automation[C]. ZhangJiaJie, China: IEEE Computer Society, 2008. 1590 - 1594.

[3] Mikko Pitkanen, Rim Moussa, et al. Erasure codes for increasing the availability of grid data storage[A]. International Conference on Internet and Web Applications and Services- AICT/ICIW' 06[C]. Guadeloupe, France: IEEE Computer Society, 2006. 1- 10.

[4] Richard S. Wilkins, Xing Du, et al. Disaster tolerant Wolfpack geo-clusters[A]. Proceedings of the 2002 IEEE International Conference on Cluster Computing[C]. Chicago, USA: IEEE Computer Society 2002, 12. 1- 6.

[5] Yanlong Wang, PZhanhuai Li, et al. RWAR: A resilient Wirr dow-consistent asynchronous replication protocol[A]. Proceedings of the The SecO International Conference Availability, Reliability and Security[C]. Vienna, Austria: IEEE Computer Society, 2007. 499- 505.

[6] Jun Feng, Lingling Cui, et al. Toward seamless grid data access: design and implementation of GridFTP on .NET[A]. The 6th IEEE/ACM International Workshop[C]. Vienna University of Technology, Austria: IEEE Computer Society, 2005. 1- 8.

[7] Sudharshan, Vazhkudai. Enabling the co-allocation of Grid Data transfers[A]. Proceedings of the Fourth International Workshop on Grid Computing[C]. Phoenix, Arizona, USA: IEEE Conrputer Society, 2003. 1- 8.

[8] R. S. Bhuvaneswaran, Yoshiaki Katayama, et al. Dynamic co-allocation scheme for parallel data transfer in Grid environment[A]. Proceedings of the First International Conference on Semantics, Knowledge, and Grid[C]. Beijing, China: IEEE Conrputer Society, 2006. 1- 6.

[9] William Allcock, John Bresnahan, et al. The globus striped

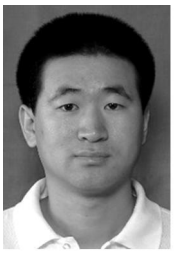
© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

GridFTP framework and server[A]. Proceedings of the 2005 ACM/IEEE SC|05 Conference[C]. Seattle, WA, USA: IEEE Computer Society, 2005. 1-11.

[10] Sudharshan, Vazhkudai. Distributed Downloads of Bulk, Replicated Grid Data[J]. Journal of Grid Computing, 2004, 2(1): 31-42.

[11] Gaurav Khanna, Umit Catalyurek, et al. A dynamic scheduling approach for coordinated wide-area data transfers using GridFTP[A]. Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium [C]. Miami, Florida, USA: IEEE Computer Society, 2008. 1-12.

作者简介:

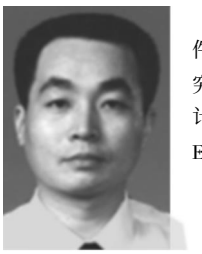


曲明成 男, 1980 年生于黑龙江省哈尔滨市. 哈尔滨工业大学计算机科学与技术学院博士研究生. 主要研究方向为网格计算、嵌入式计算、空间计算、企业智能计算、实时嵌入式操作系统.

E-mail: qumingcheng@126.com

吴翔虎 男, 1968 年生于黑龙江哈尔滨市. 哈尔滨工业大学计算机科学与技术学院教授, CCF 高级会员. 主要研究方向为实时与嵌入式计算、网格计算、空间计算、实时嵌入式操作系统、汽车计算等.

E-mail: Wuxianghu@hit.edu.cn



廖明宏 男, 1966 年生于台湾. 厦门大学软件学院教授, 博士生导师, CCF 高级会员. 主要研究领域为人工智能、嵌入式计算、普适计算、网格计算、无线传感器网络等.

E-mail: liao@xmu.edu.cn

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net