

信息知识库中的数据可用性恢复策略*

徐 鹏¹, 孟宇龙², 朱 群¹, 侍守创¹, 龚玉婷¹

(1. 中国船舶重工集团公司第七一六研究所 江苏杰瑞科技集团有限责任公司, 江苏 连云港 222002;
2. 哈尔滨工程大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

摘 要:信息知识库在工业生产中能够提高生产效率、降低资源消耗,但数据失效频率高。针对多节点失效重构的纠删码策略较少,且未充分考虑失效数据重构时各参与节点间的链路关系,导致重构效率较低。针对信息知识库数据失效,提出一种新的重构策略。根据节点的数据处理能力,选取数据处理能力最高的新生节点为路由节点;依据路由节点与候选供应节点及剩余空闲节点的链路带宽,确定供应节点及新生节点,从而构建数据重构网络拓扑,提高失效数据重构效率。实验结果证实,与传统纠删码策略性能比较,该方法有较短的重构时间和较高的重构成功率。

关键词:信息知识库;纠删码;多节点失效;数据处理;链路带宽

中图分类号:TN95 文献标志码:A 文章编号:1001-2486(2021)06-102-06

Data availability recovery strategy in information knowledge base

XU Peng¹, MENG Yulong², ZHU Qun¹, SHI Shouchuang¹, GONG Yuting¹

(1. Jiangsu JARI Technology Group Co. Ltd., The 716th Research Institute of CSIC, Lianyungang 222002, China;
2. College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: The information knowledge base can improve production efficiency and reduce resource consumption in industrial production, but it frequently meets with data failure. Furthermore, erasure coding strategies for multi-node failure reconstruction are limited, and the link relationship between participating nodes during the reconstruction of failed data has not been fully considered, resulting in low reconstruction efficiency. Aiming at the data failure of the information knowledge base, a multi-node failure reconstruction method was proposed. According to the data processing capacity of the node, the new node with the highest data processing capacity was selected as the routing node; then according to the link bandwidth of the routing node, the candidate supply node and the remaining idle nodes, the supply node and the new node were determined, thereby constructing the data reconstruction network topology and improving the efficiency of reconstruct failed data. Experimental results show that this method has shorter reconstruction time and higher reconstruction success rate when compared with traditional erasure code reconstruction methods.

Keywords: information knowledge base; erasure code; multi-node failure; data processing; link bandwidth

相比于传统数据库,信息知识库不仅包含了大量的数据,还包含了规则和过程性知识,信息化地从知识库中提取数据,能够有效提高生产效率^[1]。由于数据的不断积累及爆发式的增加,知识库的存储节点数量也随之不断膨胀。为控制成本,大规模的知识库存储系统的搭建一般选用廉价的存储服务器,当设备离线或故障时,存储节点失效,数据丢失^[2],故保证知识库数据的可靠性和完整性至关重要。

现有保证数据可靠性及完整性的方式主要包括副本冗余^[3]与纠删码^[4],副本冗余技术成本随

数据量的增大而不断提高,纠删码技术可以在存储成本低于副本冗余技术的前提下,获得相同或更高的数据可靠性及完整性^[5-6],但其失效数据重构过程的数据传输会造成较多的网络资源消耗,且重构速度有待提高。如何提高纠删码的失效数据重构性能成为研究热点。传统纠删码数据重构采用供应节点与新生节点数据直接传输的方式,该方式瓶颈取决于供应节点与新生节点间网络状况最差的一条链路,对整体重构性能影响严重。针对该问题,Li等^[7-8]以树形拓扑结构代替节点直连方式,该方式将新生节点作为重构拓扑

* 收稿日期:2020-06-24
基金项目:科技部智能工厂与网络协同制造专项资助项目(2020YFB1712600);工信部高技术船舶资助项目(MC-201720-Z02);工信部船舶总装建造智能化专项资助项目(CJ04N20)
作者简介:徐鹏(1982—),男,山东威海人,研究员,博士,硕士生导师, E-mail:xupeng716@126.com

的根节点,参与重构的供应节点作为叶节点,叶节点数据经过计算处理后上传至父节点。当上级节点获取到数据后,与本节点数据按照算法进行计算,然后向其父节点发送计算结果,依照该方法递归至根节点,重构过程结束。树形重构方法虽然一定程度上降低了传输链路瓶颈对重构效率的影响,但是数据完整性较差。

在实际生产环境中,当有单个存储节点失效时,系统并不会立即进行数据恢复,而是在达到指定失效节点数量或设定时间时开始重构^[9-10],多节点失效情况下星形及树形重构方式性能下降严重。基于多节点失效情形,本文提出链路带宽和路由节点的重构方法 (Reconstruction Method based on Link Bandwidth and Routing Node, RMLBRN),首先根据节点的数据处理能力从候选新生节点中选举出路由节点,然后根据路由节点与候选供应节点间链路带宽选出供应节点,根据路由节点与候选新生节点间带宽确定新生节点,从而构成失效数据重构的网络拓扑。在路由节点接收供应节点数据重构出失效数据后,发送至新生节点,完成数据重构,从而有效降低重构时间。

1 研究现状

文献[11-12]提出了 Regenerating Codes,该方法不仅满足了最大可分离码的性质,而且引入了网络编码来优化重构带宽。在重构过程中,选取尽可能多的节点参与重构,降低了重构过程中带宽的消耗。但该方法限制每个供应节点必须向新生节点传输等量数据,使得新生节点可选择使用的链路受限,无法利用其中具有较高带宽的链路资源,从而重构时间较长。文献[13]基于树形重构方法对节点间带宽进行排序,并优先选取可用带宽较好的节点参与重构过程,降低了数据传输对重构性能的影响,但该重构方法为贪心策略,时空复杂度会随存储系统节点增多而不断提高。文献[14]通过避免使用上一次重构时带宽最小的链路来改善重构时间,但该方法性能随重构次数增加而下降,且在多节点失效情况下性能下降更加明显。

针对存储系统中存在多个失效节点的问题,文献[15]中将纠删码参与重构节点的数据划分至不同的块中,重构过程中各节点根据所在块完成相应任务,相较单节点重构有更高的重构速率,但该重构策略时间复杂度较高;基于星形结构的串行修复策略^[16] (Star Structure based serial Repair, SSR)与基于树形结构的串行修复策略^[17]

(Tree structure based serial Repair, TSR)采用串行方式重构失效数据,重构时间较长且占用网络资源较多;文献[18]针对现有多节点重构方案未考虑新生节点间的数据传输问题,提出了基于带宽的节点选择策略 (Bandwidth based Weak and Strong Judgement, B-WSJ),考虑新生节点间信息传输,可以降低重构时间,但仍为串行重构,且在判断新生节点间数据传输时会增加重构时间;文献[19]从节点间网络距离角度出发,通过统计各供应节点与新生节点间的网络距离,找出网络距离总和最小的链路进行重构,以此降低重构时间,但该方法并未考虑实时带宽及链路情况变化。

2 问题描述

一般以 (n,k,d) -纠删码代表将数据对象划分为 k 个大小完全一致的数据块,通过对该 k 个数据块进行纠删码编码生成 $n(n>k)$ 个大小相同的编码块的纠删码,当有任意 $d(d\geq k)$ 个编码块存活时,通过解码运算即可恢复失效数据或原有数据。以 $(6,3,3)$ -纠删码为例,首先将数据对象分为3个数据量完全一致的数据块 $B_1、B_2、B_3$,根据编码规则对3个数据块进行编码生成 $C_1、C_2、C_3$ 3个编码块,大小与数据块相同。分别将每个块存放于不同的节点,根据最大可分离性质,当系统中有少于等于3个失效块时,便可恢复出失效数据。划分及编码过程如图1所示。

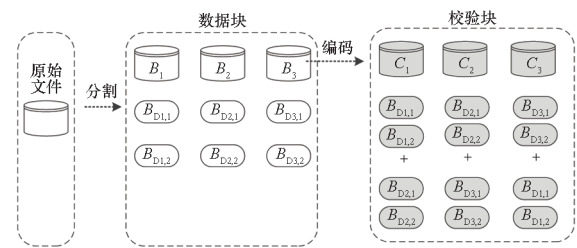


图1 $(6,3,3)$ -纠删码

Fig. 1 $(6,3,3)$ -erasure code

假设系统设定当存在两个失效节点时,开始重构失效数据。以 SSR 方法与 TSR 方法为例,当系统进行重构时,会在存活的空闲节点中随机选取2个供应节点作为新生节点,存活的数据节点和编码节点向其传输该节点的所有数据并进行重构。传统重构方法在选择新生节点时并未考虑与供应节点间的链路带宽,性能上有较大瓶颈;在重构时每个节点传输该节点的所有数据,数据冗余较大,占用了较多的网络资源,资源消耗严重。

针对多节点失效情况,本文提出了 RMLBRN 算法。该方法首先根据新生节点的数据处理能力,选择出负责接收、计算并传输数据的路由节点;然后选取与路由节点可用带宽较大的供应节点及新生节点参与重构过程,生成最大可用带宽重构拓扑,来提高多节点失效时的重构效率。

3 算法及分析

3.1 选举路由节点

由于系统中各节点性能存在差异,磁盘的 I/O、内核数目、内存型号、芯片型号、硬盘的缓存和转速以及 CPU 的主频等都有可能是导致节点能力异构的因素。所有节点不可能一成不变,在经过一段时间的使用之后,节点的一些性能也会随时间而改变,需通过及时更新节点数据处理能力来保证实验参数的准确性。在分析过后,本节以以下衡量标准定义节点的数据处理能力:首先根据节点的主要决定因素初始化节点的数据处理能力,然后根据每次系统的设定改变,或者以一次完整重构时间为基数,经加权计算后更新该节点性能。

以 (n, k, d) - 纠删码为例,当系统存在 $r(r \leq n - d)$ 个失效节点时,开始重构失效数据。假设该系统共包含 N 个存储节点,其中 $n - r$ 个供应节点需进行数据读取、编码及上报,节点压力较大,故在剩余 $N - n$ 个节点中选择 r 个空闲节点作为新生节点。

首先选取磁盘 I/O、CPU 核数、主频、内存大小等作为衡量节点数据处理能力的标准。分别用 α 表示节点的磁盘 I/O, β 表示 CPU 核数, γ 表示节点主频, θ 表示节点内存,同时以 $c_\alpha, c_\beta, c_\gamma, c_\theta$ 表示四项标准的加权参数,并满足 $c_\alpha + c_\beta + c_\gamma + c_\theta = 1$, 则节点 i 的数据处理能力可表示为:

$$S_i = c_\alpha \alpha + c_\beta \beta + c_\gamma \gamma + c_\theta \theta \quad (1)$$

假设节点 i 在重构过程中参与的数据读取、编码等操作的总计算量为 P_i , 则节点 i 参与重构的总时间为:

$$t = T(P_i / S_i) \quad (2)$$

其中函数 $T(\cdot)$ 将计算结果转换为时间单位(s), 以便后续处理, 其转换结果与节点数据处理能力成反比。

针对节点状态的变化对数据处理能力的影响, 在式(2)初始化节点的数据处理能力后, 当达到设定的时间阈值或数据重构后, 更新节点的数据处理能力。以 $n_{i,t}$ 表示节点 i 本次参与重构消耗的时间,

$l_{i,t}$ 表示节点 i 上一次参与重构消耗的时间, 单位均为 s, 则更新节点 i 的参与重构时间为:

$$t_i = a l_{i,t} + (1 - a) n_{i,t} \quad (3)$$

式中, $a(0 \leq a \leq 1)$ 为上次参与重构时间的权重。通过式(3), 节点 i 的数据处理能力与重构时间建立关系, 并成反比。当路由节点数据处理能力越强时, 系统重构效率越高。

3.2 选择参与重构的节点

在选择参与重构的供应节点时, 首先需要获取存活节点与新生节点间的可用带宽大小, 然后按照最大生成树原理, 依次选取与新生节点有最大可用带宽的存活节点作为供应节点, 直到供应节点数量满足所选纠删码策略的重构要求为止。

由于测量各节点间链路带宽仅需 2 ~ 4 个往返时延^[20], 相对重构过程影响可忽略, 且短时间内带宽波动较小, 因此在数据重构期间可将带宽设置为固定值^[21-22]。假设各参与重构节点的数据传输量均为 d 。具体供应节点选择步骤如下。

以 (n, k, d) - 纠删码为例, 重构阈值为 $r(r \leq n - d)$, 引入供应节点集合 C_c 及新生节点集合 C_n , 用以存放参与失效数据重构的供应节点及新生节点。首先从空闲新生节点中根据节点的数据处理性能选择路由节点 N_c , 放入集合 C_n 。

引入边集 $E_c = \{e_0, e_1, \dots, e_{n-r-1}\}$ 表示路由节点与候选供应节点间的链路, 其值代表数据传输的网络带宽, 值为 0 表示节点间无链路连接。算法根据路由节点与候选供应节点的边集 E_c 依次选择具有链路带宽的节点, 确定为供应节点, 将该点存入集合 C_c 中并从边集中删去该链路, 根据此方法选出共 d 个供应节点截止。

供应节点确定后, 需在空闲节点中选择剩余 $r - 1$ 个新生节点。引入边集 $E_n = \{e_0, e_1, \dots, e_{N-n-2}\}$ 表示路由节点与候选新生节点间的链路, 其值代表数据传输的网络带宽, 值为 0 表示节点间无链路连接。算法根据路由节点与候选节点的边集 E_n 依次选择具有链路带宽的节点, 确定为新生节点, 将该点存入集合 C_n 中并从边集中删去该链路, 根据此方法选出共 $r - 1$ 个新生节点截止, 从而构成以路由节点为中心的数据重构网络拓扑。参与重构的节点选择算法如算法 1 所示。

以 $(6, 3, 3)$ - 纠删码为例, 重构阈值为 2, 介绍 RMLBRN 方法的具体过程。节点间网络链路及带宽如图 2 所示, 节点 1 ~ 4 为存活节点, 节点 5 ~ 6 为失效节点, 节点 7 ~ 10 为空闲节点。当系统中存在 2 个失效节点时, 开始数据重构过程。

算法 1 参与重构节点选择算法

Alg. 1 Selection algorithm of nodes participating in reconstruction

输入: C_c, C_n, E_c, E_n, N_c
输出: 数据重构网络拓扑

1. 初始化 $i, j, k, max \leftarrow 0$;
2. for $i \leftarrow 0$ to $n - r - 1$ do
3. for $j \leftarrow 0$ to $n - r - i - 2$ do
4. if $e_j > e_{j+1}$ in E_c then
5. $max = e_{j+1}$
6. $e_{j+1} = e_j$
7. $e_j = max$
8. end if
9. end for
10. end for
11. for $i \leftarrow n - r - 1$ to $n - r - 1 - d$ do
12. add the node of $E_c[i]$ to C_c
13. end for
14. for $i \leftarrow 0$ to $N - n - 2$ do
15. for $j \leftarrow 0$ to $N - n - i - 3$ do
16. if $e_j > e_{j+1}$ in E_n then
17. $max = e_{j+1}$
18. $e_{j+1} = e_j$
19. $e_j = max$
20. end if
21. end for
22. end for
23. for $i \leftarrow N - n - 2$ to $N - n - r - 3$ do
24. add the node of $E_n[i]$ to C_n
25. end for

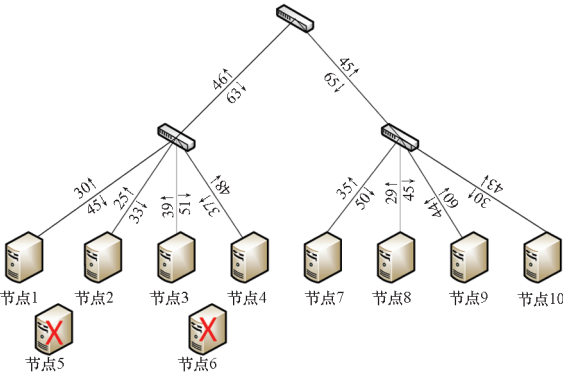


图 2 (6,3,3) - 纠删码节点连接图

Fig. 2 Connection graph of the nodes of (6,3,3) - erasure code

首先根据节点的数据处理能力在空闲节点中选出节点 9 作为路由节点;然后根据路由节点与候选供应节点间的网络带宽,依次选出节点 3、节

点 4、节点 1 作为供应节点;最后根据路由节点与候选新生节点间的网络带宽,选择出节点 8 作为新生节点,从而构成数据重构的网络拓扑,如图 3 所示。

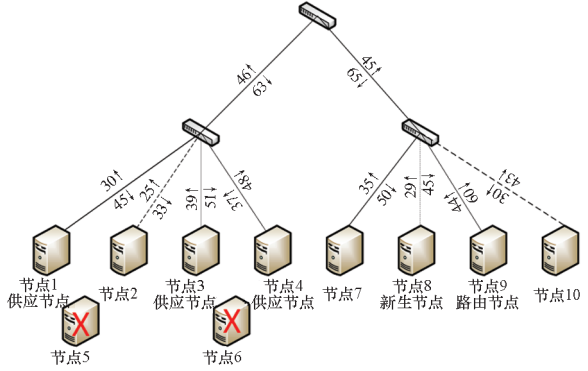


图 3 (6,3,3) - 纠删码数据重构网络拓扑

Fig. 3 Data reconstruct network topology of (6,3,3) - erasure code

4 实验验证及分析

本节进行不同带宽及不同失效节点个数下, RMLBRN、TSR 和 B-WSJ 重构方法的性能对比实验。

4.1 实验目的和指标

实验将 RMLBRN 与现有存储系统中常用的 TSR 和 B-WSJ 失效重构方法对比,通过数据重构时间、失效节点变化对重构时间的影响及重构成功率实验观察不同方法的性能,三种方法均采用块存储方式。数据重构时间,指重构过程开始到所有失效节点全部重构完成所经历的时间长度,是验证性能的最直接和最主要的指标;重构成功率,指在重构过程开始到重构结束时,失效块被成功完成重构的概率。在重构过程中,可能会出现供应节点或新生节点失效的情况,导致重构失败,所以重构成功率一般由成功重构的失效块数量除以所有失效块的数量。

4.2 实验环境

实验基于 HDFS-RAID 平台,该平台被 Facebook 用于解决 HDFS 使用多副本策略导致存储成本较大的问题,实验采用了纠删码策略。实验设置 10 个 DataNode、1 个 RaidNode 及 1 个 NameNode,其中 DataNode 用于存放数据, RaidNode 用于对数据块的编码及失效数据的重构, NameNode 用于管理系统中存放的数据。实验模拟生成 64 G 数据,并划分为 64 MB 的数据块,通过编码后存放于各节点中。各节点间通过

10 GB/s 的万兆网交换机连接。节点失效或离线通过随机断开节点的方式进行模拟。

实验选取节点的 I/O、CPU 核数、主频、内存等作为衡量节点的数据处理能力,并模拟不同数值及相应权重。磁盘 I/O 由于占用时间较多,故分配权重为 0.4;CPU 核数取值范围为 1~2 个,占比 0.3;主频取值范围为 0.5~2.3 GHz,占比 0.1;内存取值范围为 $1 \times 32 \text{ GB} \sim 8 \times 32 \text{ GB}$,占比 0.2。

4.3 带宽对重构时间的影响

首先探究网络带宽的变化对 RMLBRN、TSR 和 B-WSJ 方法重构时间的影响。其中 TSR 方法为串行重构方法,所有参与重构的供应节点将数据按树形拓扑结构传输给新生节点进行数据重构;B-WSJ 算法则在多节点失效情况下考虑节点间的最大可用带宽进行串行重构。以 RS(10,5) 编码策略为例,系统存在 4 个失效节点时开始重构。试验带宽服从均匀分布,范围为 1~100 Mbit/s,实验结果采用 50 次实验数据的平均值。实验通过设置不同的网络条件来进行对比,实验结果如图 4 所示。

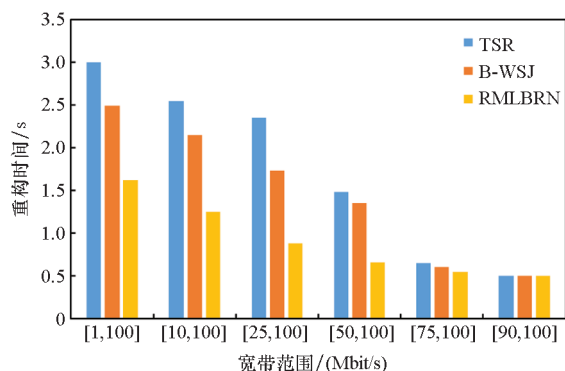


图 4 带宽分布对重构时间的影响

Fig.4 Influence of bandwidth distribution on reconstruction time

实验结果表明,当带宽波动较大时,各方法的性能均有大幅下滑。当带宽较高而且平稳时,RMLBRN 重构算法的性能比 B-WSJ 重构算法提高了 12.44%,比 TSR 重构算法性能提高了 17.36%。当带宽分布为 $[1,100]$ 时,RMLBRN 重构方法较 B-WSJ 和 TSR 重构方法性能分别提高 36.01% 和 46.67%。

4.4 失效节点数对重构时间的影响

实验在 RS(10,5) 编码策略下,以随机断开节点的方式模拟不同数量的失效节点。为降低带宽对重构时间的影响,固定带宽范围为 90~100 Mbit/s;并根据编码的重构阈值,设定失效节点数为 1~4。实验结果如图 5 所示。

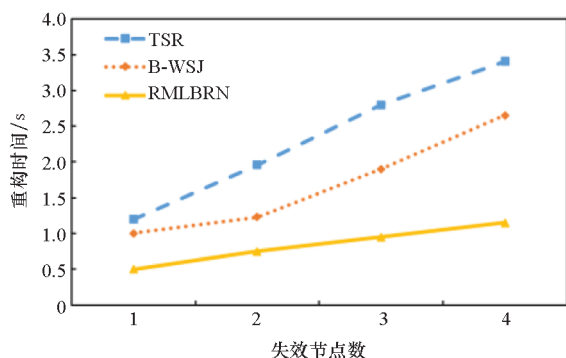


图 5 失效节点数对重构时间的影响

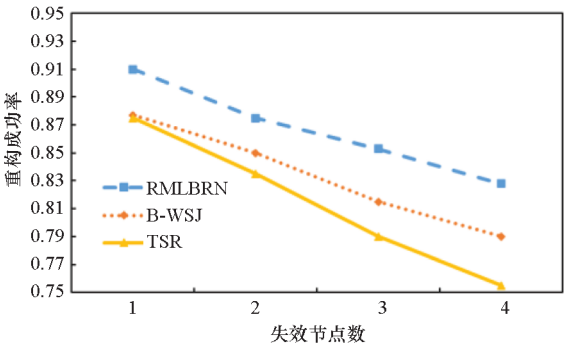
Fig.5 Influence of the number of failed nodes on reconstruction time

从图 5 可知,当失效节点数由 1 增加到 4 时,B-WSJ 和 TSR 方法的性能下降较为明显,RMLBRN 方法性能较为稳定。由于 TSR 为串行重构,虽然简单易实现,但是针对多节点失效情况串行效率较低;B-WSJ 方法虽然重构过程考虑了节点间的可用带宽,但该方法基于贪心策略,当失效节点增多时性能会随之下降。而 RMLBRN 算法通过选取数据处理能力最强的新生节点作为路由节点,能够提高计算的处理能力,而且选取与新生节点有最大可用带宽的存活节点作为供应节点来生成最大重构树,进行数据重构,相比 TSR 方法降低了串行重构的资源消耗,进一步提高了重构效率。

4.5 失效节点数对重构成功率的影响

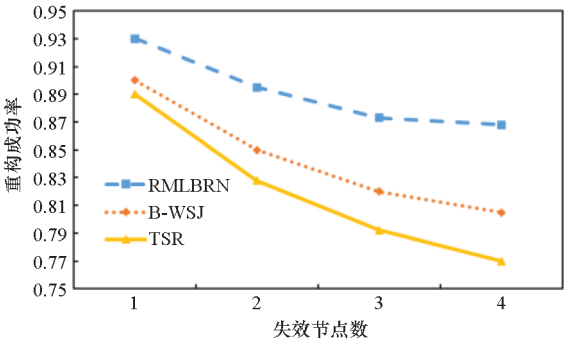
最后验证不同失效节点数对各重构方法重构成功率的影响。为验证普遍,分别以 RS(10,5) 和 RS(12,6) 编码策略为例进行实验(也可根据具体环境选择其他 RS 码),固定带宽分布为 90~100 Mbit/s,降低带宽因素对重构成功率的影响,并设定失效节点数为 1~4,实验结果如图 6 所示。

从图 6 可以看出,当失效节点数量上升时,各方法的重构成功率都出现不同程度的下降。但在相同失效节点数的情况下,RMLBRN 方法根据路由节点与供应节点、新生节点间的最大链路带宽建立数据重构拓扑,降低了失效数据重构时间,使得节点失效概率下降,从而提高了失效节点的重构成功率。同时可以看出,B-WSJ 和 TSR 方法重构成功率下滑程度较大,RMLBRN 方法下滑幅度相对较小且一直高于 B-WSJ 和 TSR 方法。与 RS(10,5) 编码策略相似,RS(12,6) 编码策略下三种重构方法的重构成功率也均有所下滑。但因为 RMLBRN 方法通过树形传输方式,规避了带宽较小的链路,提高了重构速度,因此成功率下滑较小。而且,针对数据分块较多的纠删码策略,



(a) RS(10,5)编码策略下失效节点数对重构成功率的影响

(a) Influence of the number of failed nodes on the success rate of reconstruction under the RS(10,5)



(b) RS(12,6)编码策略下失效节点数对重构成功率的影响

(b) Influence of the number of failed nodes on the success rate of reconstruction under the RS(12,6)

图 6 失效节点数对重构成功率的影响

Fig. 6 Influence of the number of failed nodes on the success rate of reconstruction

RMLBRN 方法的属性传输策略能够较为有效地提高重构效率,能够降低重构时参与重构的数据块失效的影响,提高重构成功率。

5 结论

随着数据量的急速膨胀,信息知识库存储节点失效时有发生,利用纠删码技术,能够在降低存储成本的同时,达到与副本冗余技术相同甚至更高的数据可用性及可靠性。但传统纠删码技术多为单点重构,而实际生产环境中多节点失效情况更为普遍。本文针对多节点失效情境,提出了一种 RMLBRN 重构方法,通过存储节点的数据处理能力选举出路由节点,并根据路由节点与其他节点间链路带宽选择出供应节点及新生节点,从而形成数据重构拓扑,在很大程度上降低了多节点失效的重构时间,且具有较高的重构成功率。实验表明,RMLBRN 方法重构性能较优。本文提出

的 RMLBRN 方法可能会构造出多个不同的数据重构拓扑,如何使网络拓扑的重构时间最短是后续研究的方向。

参考文献 (References)

[1] 管官, 林焰, 纪卓尚. 基于知识的船体结构快速设计及优化[J]. 船舶力学, 2017, 21(4): 472-483.
GUAN Guan, LIN Yan, JI Zhuoshang. Knowledge-based quick design and optimization for hull structure[J]. Journal of Ship Mechanics, 2017, 21(4): 472-483. (in Chinese)

[2] OUYANG J, LIN S D, JIANG S, et al. SDF: software-defined flash for web-scale Internet storage systems [C]// Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, 2014: 471-484.

[3] WANG Y J, LI S J. Research and performance evaluation of data replication technology in distributed storage systems[J]. Computers & Mathematics with Applications, 2006, 51(11): 1625-1632.

[4] YE M, BARG A. Cooperative repair: constructions of optimal MDS codes for all admissible parameters [J]. IEEE Transactions on Information Theory, 2019, 65(3): 1639-1656.

[5] LIN W K, CHIU D M, LEE Y B. Erasure code replication revisited[C]//Proceedings of Fourth International Conference on Peer-to-Peer Computing, 2004: 90-97.

[6] WEATHERSPOON H, KUBIATOWICZ J D. Erasure coding vs. replication; a quantitative comparison[C]// Proceedings of Peer-to-Peer Systems, 2002: 328-337.

[7] LI J, YANG S, WANG X, et al. Tree-structured data regeneration with network coding in distributed storage systems[C]// Proceedings of 17th International Workshop on Quality of Service, 2009: 1-9.

[8] LI J, YANG S, WANG X, et al. Tree-structured data regeneration in distributed storage systems with regenerating codes[C]// Proceedings of IEEE INFOCOM, 2010: 1-9.

[9] LI R H, LI X T, LEE P P C, et al. Repair pipelining for erasure-coded storage [C]// Proceedings of USENIX Technical Conference, 2017: 567-579.

[10] 谭子军, 何连跃. 分布式存储系统中用户磁盘空间分配策略[J]. 计算机工程, 2010, 36(9): 44-46.
TAN Zijun, HE Lianye. User disk space allocation policy in distributed storage system[J]. Computer Engineering, 2010, 36(9): 44-46. (in Chinese)

[11] DIMAKIS A G, GODFREY P B, WU Y N, et al. Network coding for distributed storage systems[J]. IEEE Transactions on Information Theory, 2010, 56(9): 4539-4551.

[12] YU Q, SUNG C W, CHAN T H. Irregular fractional repetition code optimization for heterogeneous cloud storage[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(5): 1048-1060.

[13] GONG Q Y, WANG J Q, WEI D S, et al. Optimal node selection for data regeneration in heterogeneous distributed storage systems [C]// Proceedings of 44th International Conference on Parallel Processing, 2015: 390-399.

- [10] SURACE C, RUOTOLO R, STORER D M. Detecting nonlinear behaviour using the volterra series to assess damage in beam-like structures [J]. *Journal of Theoretical and Applied Mechanics*, 2011, 49(3): 905–926.
- [11] CHATTERJEE A. Nonlinear dynamics and damage assessment of a cantilever beam with breathing edge crack[J]. *Journal of Vibration and Acoustics*, 2011, 133(5): 051004.
- [12] ANDREAUS U, CASINI P, VESTRONI F. Non-linear dynamics of a cracked cantilever beam under harmonic excitation [J]. *International Journal of Non-Linear Mechanics*, 2007, 42(3): 566–575.
- [13] CHENG S M, SWAMIDAS A S J, WU X J, et al. Vibrational response of a beam with a breathing crack[J]. *Journal of Sound and Vibration*, 1999, 225(1): 201–208.
- [14] LONG H, LIU Y L, LIU K F. Nonlinear vibration analysis of a beam with a breathing crack[J]. *Applied Sciences*, 2019, 9(18): 3874.
- [15] MA H, ZENG J, LANG Z Q, et al. Analysis of the dynamic characteristics of a slant-cracked cantilever beam [J]. *Mechanical Systems and Signal Processing*, 2016, 75(15): 261–279.
- [16] BRODA D, PIECZONKA L, HIWARKAR V, et al. Generation of higher harmonics in longitudinal vibration of beams with breathing cracks [J]. *Journal of Sound and Vibration*, 2016, 381(27): 206–219.
- [17] PUGNO N, SURACE C, RUOTOLO R. Evaluation of the non-linear dynamic response to harmonic excitation of a beam with several breathing cracks [J]. *Journal of Sound and Vibration*, 2000, 235(5): 749–762.
- [18] BOVSUNOVSKY A, SURACE C. Non-linearities in the vibrations of elastic structures with a closing crack: a state of the art review [J]. *Mechanical Systems and Signal Processing*, 2015, 62/63: 129–148.
- [19] DIMAROGONAS A D, PAIPETIS S A, CHONDROS T G. *Analytical methods in rotor dynamics* [M]. Dordrecht: Springer Netherlands, 2013.

(上接第 107 页)

- [14] SUN W D, WANG Y J, PEI X Q. Tree-structured parallel regeneration for multiple data losses in distributed storage systems based on erasure codes[J]. *China Communications*, 2013, 10(4): 113–125.
- [15] SHUM K W. Cooperative regenerating codes for distributed storage systems [C]// *Proceedings of IEEE International Conference on Communications (ICC)*, 2011: 1–5.
- [16] PEI X Q, WANG Y J, MA X K, et al. Cooperative repair based on tree structure for multiple failures in distributed storage systems with regenerating codes[C]//*Proceedings of the 12th ACM International Conference on Computing Frontiers*, 2015: 784–792.
- [17] NGUYEN B, TAN H, DAVIS K, et al. Persistent octrees for parallel mesh refinement through non-volatile byte-addressable memory[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2019, 30(3): 677–691.
- [18] 刘佩, 蒋梓逸, 曹袖. 一种基于分布式存储系统中多节点修复的节点选择算法[J]. *计算机研究与发展*, 2018, 55(7): 1557–1568.
- LIU Pei, JIANG Ziyi, CAO Xiu. Node selection algorithm during multi-nodes repair progress in distributed storage system[J]. *Journal of Computer Research and Development*, 2018, 55(7): 1557–1568. (in Chinese)
- [19] 郑力明, 李晓冬. 面向纠删码的低成本多节点失效修复方法[J]. *计算机工程*, 2017, 43(7): 110–118.
- ZHENG Liming, LI Xiaodong. Low-cost multi-node failure repair method for erasure codes[J]. *Computer Engineering*, 2017, 43(7): 110–118. (in Chinese)
- [20] LAM I, SZE BENI S, SZILAGYI G, et al. Wuala [EB/OL]. (2016–02–13) [2020–05–24]. <https://tresorit.com/business/wuala-alternative>.
- [21] ZHANG H Y, LI H, SHUM K W, et al. Concurrent regenerating codes[J]. *IET Communications*, 2017, 11(3): 362–369.
- [22] HU Y P, LIU Y H, LI W J, et al. Unequal failure protection coding technique for distributed cloud storage systems[J]. *IEEE Transactions on Cloud Computing*, 2021, 9(1): 386–400.