网络编码系统中基于访问频度的数据重建方法*

李 凯
(暨南大学 计算机科学系，广东 广州 510632)

摘 要：在分布式存储系统中，通常需要在节点失效之后引入新节点并重建数据，以保证系统的可用性。网络编码(Network Coding)存储技术通过数据在存活节点内部作线性组合，可以大幅度降低数据重建时的下载带宽，因此近网络编码技术在节点修复过程中具有非常重要的地位。但同时其大量的线性组合运算也导致了相当可观的时间开销，极大地影响了数据重建的效率和用户的响应请求。基于网络编码文件系统(NCFS)，提出了一种结合 80-20 法则的数据重建方法，并作出了程序实现与仿真验证。实验结果表明，新系统在重建效率、用户平均响应时间及吞吐率方面均有较大提升。

关键词：网络编码；NCFS；数据重建；80-20 法则
中图分类号：TP311.5 **文献标识码**：A **文章编号**：1674-7720(2014)06-0007-03

A frequency-based data rebuilding method in network coding file system

Li Kai
(Department of Computer Science, Jinan University, Guangzhou 510632, China)

Abstract: In order to keep data avialablity, it is necessary to introdue new disk and rebuild data after disk failure. Network coding have been playing a more and more important role in the data rebuild process for it's advanced feature that it can achieve optimal download bandwidth during rebuild process by making linear combinations inside the surviving disk node. In the meantime, the combination operations also bring a mount of time comsumption, which greatly lower rebuild efficiency and increase data request time. We promote a new rebuild method based on 80/20 rule in Network Coding File System (NCFS). Implementation and simulation are given. Experiment result shows that new system makes great performance improvement in rebuild time, response time and throughput.

Key words: network coding; NCFS; disk rebuild; 80-20 rule

在这个大数据的年代，数据量增长的速度是惊人的。据 IDC 报告显示，预计到 2020 年全球数据总量将超过 40 ZB(相当于 4 万亿 GB)[1]，这一数据量是 2011 年的 22 倍。为了给海量数据提供有效的存储及服务的能力，诞生了许多大规模数据存储系统，比如 GFS、Hadoop、OceanStore、Lustre、Gluster 等。在这些大型存储系统中，数据分布在一系列的节点(磁盘等物理介质)上，为了保证数据的可用性，系统必须能够容忍节点失效。为了达到这一目的，分布式存储系统引入了冗余数据以提供容错能力。

一般的容错技术包括副本技术，纠删码技术和网络编码技术。副本技术对一个数据对象创建多个副本，并将这些副本分散到不同的节点上。当一个节点失效时，可以通过访问其他节点的数据副本来重建新节点。比如 GFS[1]为每个数据块提供了 3 个副本。纠删码技术是能够容忍一个或多个节点同时失效的编码技术，而且比副本技术有更高的空间存储效率。常见的纠删码有 Reed-Solomon 码、LDPC 码等。网络编码技术通过选择特殊的编码系数来构造生成矩阵，在节点修复时，把存储在同一节点上的若干数据块做线性运算，所以该节点传输一个数据块就等于提供了做运算之前的若干个数据块的信息，从而有效地节省了带宽。

DIMAKIS 等人于 2007 年首先在分布式存储系统中引入网络编码思想，提出了一种称为再生码(regenerating code)[2]的编码技术。随后，Rashmi 等人提出了 E-MBR (Exact minimum Bandwidth Regenerating)码[3]，突破了网络编码的理论阶段，给出了一个具体的最优带宽再生码方案。虽然网络编码在数据重建时，下载带宽方面表现优越，但是其付出的运算开销却不可忽视[2]。据 NCFS[4]研究表

* 基金项目：暨南大学研究生“菁英学子”计划项目资助(201311)

明，网络编码在退化模式下的表现明显不如 RAID5 和 RAID6。Tian Lei 等人实现了以访问频度优先的数据重构优化方法[5]来改善磁盘阵列中数据重建缓慢的问题，不过他们只限于对 RAID5 和 RAID10 的研究。基于此，本文提出了一种在网络编码修复过程中利用 80/20 法则来加快数据重建过程的方法。在 NCFS 平台上进行了仿真实现，并对 RAID5、RAID6 和 E-MBR 编码在节点重建时间、用户平均响应时间和吞吐率方面进行了比较。

1 背景知识

1.1 帕雷托法则(Pareto principle)

帕雷托法则又称 80-20 法则，在计算机科学里，80-20 法则代表 80%的资源只被 20%的操作所使用。具体到文件系统的访问行为，是指 80%的请求往往集中在 20%的文件上，从而导致某一部分数据被频繁重复地访问，而其他数据则相对访问频度较低。比如视频网站约 20%的视频文件由于经常被观看而必须保存在内存中，以提供快速及时的响应；而剩余的 80%视频文件则观看次数较少，可把这些数据置于存储后端，需要访问时再从后端提取。

80-20 法则对数据重建具有非常重要的借鉴意义。因为一旦节点失效，系统就处于退化模式，处于退化模式下的文件系统同时兼顾重建节点和响应用户请求的工作。此时对失效节点的写请求可能被拒绝，读请求转化为对若干存活数据节点的读请求再对读出来的数据作编码运算。若 20%的热点数据迟迟没有重建完成，则会累积大量退化读写请求。此时不但额外增加了读操作和运算，而且磁盘重建数据流和用户请求数据流对不同区域的读写操作会导致磁盘来回长寻道，因而严重降低了系统的 I/O 性能和系统的响应能力。若优先重建热点数据，则能明显缩短退化模式的持续时间，改善系统的 I/O 效率和系统响应能力。

1.2 E-MBR 编码(Exact Minimum Bandwidth Regenerating Code)

一般再生码分为最小带宽再生码(MBR)和最小存储再生码(MSR)。相比 MSR，MBR 以略增加节点的存储量为代价，换取降低数据重建的带宽开销。通常用一个元组$(n,k.,m,d)$来描述一个 MBR 编码系统。数据编码后分布存储在 n 个节点上，用户连接其中任意 k 个节点可以还原出原始文件。节点修复时，新节点连接 d 个节点来完成修复。$m=n-d$，即当失效的节点多于 m 个时，就必须要通过还原整个原始文件来完成节点修复，这将带来相比常规节点修复过程高得多的带宽和计算开销。一般最基本的编码方式是 $d=n-1$。E-MBR 编码[3]是 Rashmi 等人提出来的一种准确性修复 MBR 编码。

2 实验设计

2.1 NCFS 系统架构

NCFS 是基于 FUSE[6]，实现在用户空间的网络编码文件系统。通过把物理节点挂载到当前的文件系统下面(如/mnt/ncfs)即可以像访问逻辑节点一样访问节点中的数据。NCFS 主要由文件系统层、编码层、存储层组成。文件系统层负责文件系统的操作，比如文件读、写、删除等；编码层提供了 RAID5、RAID6、E-MBR 的存储编码方式；存储层提供访问具体物理设备的接口。在实验中，用 Linux 操作系统的伪块设备/dev/loop 来模拟物理磁盘的存储行为，用户的读、写请求都是针对/dev/loop1，/dev/loop2 等块设备的读写。其系统架构如图 1 所示。

图 1 NCFS 系统架构图

2.2 以 Zipf-like 分布访问块设备

用 Zipf-like 分布模拟用户访问行为，由此产生的访问请求具有 80-20 特征。把块设备节点平均分成 n 个区域，用数组 access_rec[n]记录每个区域的访问频度，通过齐夫定律公式产生访问的块号去访问该块，访问请求完成之后修改该块号所在区域的访问频度。根据 Zipf-like 分布，可知第 i 块的访问概率为：

$$P_N(i)=\frac{\Omega}{i^{\alpha}},\text{其中 }\Omega=\left(\sum_{i=1}^{N}\frac{1}{i^{\alpha}}\right)^{-1} \quad (1)$$

其中，α 为一个常数，其取值范围是 $0<\alpha\leqslant 1$；N 为块总数，因此 Ω 也是一个常量。实验中通过齐夫定律公式产生的访问行为如表 1 所示(总访问次数为1 000 000 次，总区域数 $n=10$)。

表 1 访问频度分布

区域号	频度
0	15 060
1	19 644
2	26 106
3	183 600
4	91 713
5	55 053
6	36 667
7	549 924
8	12 320
9	9 913

各区域按访问频度排序如图 2 所示。

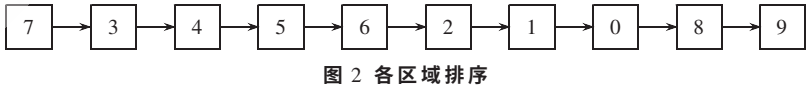

图 2 各区域排序

2.3 数据重建算法

(1)把记录访问频度的数组 access_rec[n]排序，得出从大到小记录区域号的数组 blk_seq[n]；

(2)从 blk_seq[n]中取出一个区域号，进行该区域的数据重建；

(3)重复步骤(2)，直到节点数据重建完成。

3 实验评估与分析

对新系统和原系统的平均重建时间、平均响应时间和吞吐率 3 个性能指标进行了实验数据收集，并进行了

比对。

3.1 平均重建时间

平均重建时间代表了系统的重建效率，其计算公式如下：

$$\text{平均重建时间}=\frac{\text{总重建时间}}{\text{节点数据量}} \quad (2)$$

其中总重建时间的单位为 s，节点数据量的单位为 MB，平均重建时间的单位为 s/MB。实验数据结果如图 3 所示。

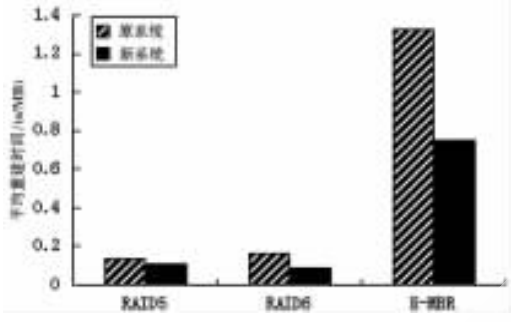

图 3 平均重建时间对比

3.2 平均响应时间

平均响应时间是指在重建过程中，系统每响应一个用户请求所用的时间。其计算公式如下：

$$\text{平均响应时间}=\frac{\text{总重建时间}}{\text{总请求数}} \quad (3)$$

实验数据结果如图 4 所示。

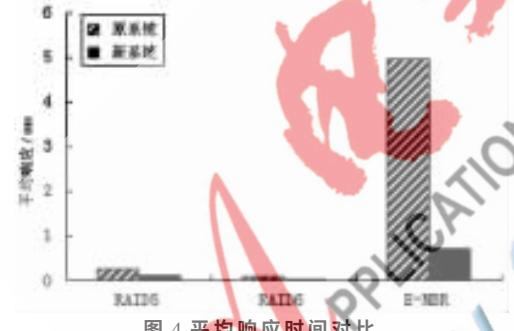

图 4 平均响应时间对比

3.3 吞吐率

吞吐率是指节点修复过程中重建数据流的每秒处理的数据量，其计算公式为：

$$\text{平均重建时间}=\frac{\text{节点数据量}}{\text{总重建时间}} \quad (4)$$

实验数据结果如图 5 所示。

实验分析：实验数据显示，E-MBR 在平均重建时间、平均响应时间和吞吐率 3 个性能指标的表现都劣于 RAID5 和 RAID6，这是因为网络编码的优势主要在于节省重建带宽，而为此付出了额外的时间开销，导致重建过程较缓慢。

不过从图表中可以看出，经过改进后的新系统在各性能的表现都比原系统好。其中平均重建时间从1.33 s/MB 降低到 0.75 s/MB，有 43.6%的改善；平均响应时间从 4.98 ms 到改进后的 0.71 ms，整整提高了 7 倍；吞吐率也

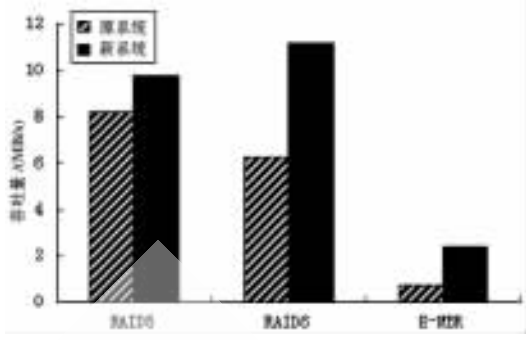

图 5 吞吐率对比

有了 3.23 倍的提升。

系统在退化模式下的性能提升关键在于让访问频度最高的数据在最短的时间里重建完成并投入服务，使对这部分数据的大量访问请求能够得到及时的响应，从而减轻了 CPU 的压力。另外，优先重建访问频度高的数据能够让重建数据流和用户请求数据流尽可能地重叠，以减少大量的磁头长寻道，从而得到更高的 I/O 效率。

本文基于网络编码文件系统(NCFS)，利用 80-20 法则对原系统的数据重建过程进行了优化，结果显示新系统在平均重建时间、平均响应时间和吞吐率各方面均有比较大的提升。实验过程中用到了伪块设备来模拟磁盘的存储行为，所有节点都在同一台计算机上。这与真实的分布式网络有一定的差别。

参考文献

[1] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system[C]. Proc. of the 19th ACM Symp. on operating Systems Principles. December, 2003:29-43.

[2] DIMAKIS A G,GODFREY P B,WAINWRIGHT M J,et al. Network coding for distributed storage systems[C]. IEEE Proc. INFOCOM, May 2007:2000-2008.

[3] RASHMI K V, SHAH N B, KUMAR P V, et al. Explicit construction of optimal exact regenerating codes for distributed storage[C]. In Proc. of Allerton Conference, 2009: 1243-1249.

[4] Hu Yuchong, Yu Chiuman, Yan Kit Li, et al. NCFS: on the practicality and extensibility of a network-coding-based distributed file system[C]. Proceedings of the 2011 International Symposium on Network Coding(NETCOD), 2011.

[5] Tian Lei,Feng Dan,Jiang Hong, et al. PRO: a popularity-based multi-threaded reconstruction optimization for RAID-Structured Storage Systems[C]. In FAST′ 07, San Jose, CA, 2007:227-290.

[6] FUSE[OL]. http://fuse.sourceforge.net/, 2013.

(收稿日期：2013-12-16)

作者简介：

李凯，男，1988 年生，硕士研究生，主要研究方向：分布式系统，网络编码。