

Cumulus: 一个基于网络编码的分布式文件系统

陈运海¹, 陈正亮¹, 王钦辉^{1,2}, 叶保留¹

(1 南京大学 软件新技术国家重点实验室, 江苏 南京 210046;

2 南京陆军指挥学院 军事训练与管理系, 江苏 南京 210045)

摘 要: 基于网络编码在并发式数据传输时能够有效地降低网络负载的优越性, 应用网络编码优化分布式文件系统的容错问题得到了广泛的关注. 为了研究网络编码在分布式文件系统中的应用及其在降低网络负载的有效性, 文中基于开源的 Hadoop 项目设计并实现了面向网络编码的分布式文件系统. 该系统通过对存储文件进行切分, 然后将这些文件切片进行编码存储, 能够以更小的冗余开销获得更好的容错性能, 同时, 能够利用网络编码的性能提高网络资源利用率.

关键词: 网络编码; 分布式文件系统; 冗余容错; Hadoop; 网络资源利用率

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2012)09-0074-05

Cumulus: A Distributed File System Based on Network Coding

CHEN Yun-hai¹, CHEN Zheng-liang¹, WANG Qin-hui^{1,2}, YE Bao-liu¹

(1 State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China;

2 Department of Military Training and Management, Army Command College, Nanjing 210045, China)

Abstract: Network coding can reduce network load in the process of concurrent data transmission, which attracts much attention on the application of network coding to optimize the fault-tolerance of distributed file system. Based on Hadoop, an open source project, we design and implement a network coding oriented distributed file system to research the application of network coding in distributed file system and its effectiveness in reducing network load. By segmenting storage files and encoding, storing the file slices, the system is able to achieve a better fault tolerance with less redundancy overhead. Meanwhile, network resource utilization can be improved by taking advantage of network coding performance.

Key words: network coding; distributed file system; redundant fault-tolerance; Hadoop; utilization of network resource

1 引言

HDFS(Hadoop Distributed File System)是 Apache 基金会下属的开源项目 Hadoop 的子项目. 作为云计算解决方案中的云存储基础架构, HDFS 采用了基于多副本的冗余容错策略. 这一策略在应对并发式数据访问有着很好的并行性, 但是, 另一方面, 随着容错等级需求的上升, 该策略所需的存储冗余开销也呈线性增长. 而起源于多播传输网络的网

络编码, 初衷是为了使得网络多播能够达到理论上的最大传输容量. 后续的研究表明, 网络编码能够明显的提升网络的吞吐量、均衡网络负载、提高带宽利用率等诸多优点^[1]. 在分布式计算的场景下, 有着广泛的研究热点.

在分布式文件系统的应用场景, 为了保证存储数据的持续可靠性, 往往需要应用某种冗余容错机制, 使得在某一存储节点出现失效下存储数据依然能够得到恢复. 同时, 为了保证该系统的容错等级,

收稿日期: 2012-02-15; 修回日期: 2012-03-23

基金项目: 国家自然科学基金青年科学基金(60903025); 江苏省科技支撑计划(BE2010179)

在数据节点失效的情况下,往往需要对失效节点所存储的数据进行修复,大规模的并发的数据传输就在所避免.利用网络编码能够有效提高带宽利用率的特性来优化这一过程的网络性能,是本文研究的重点.因此,为了研究基于网络编码的分布式文件系统的容错问题,本文基于开源的 Hadoop 项目设计并实现了 Cumulus 文件系统,一个基于网络编码容错的分布式文件系统.

2 相关技术介绍

2.1 HDFS 中基于多副本的容错存储策略

当 HDFS 系统中的副本扫描进程发现某些副本失效时,就会启动修复进程,该进程选取完好的副本并对其进行复制,放置在完好的数据节点上并向名字节点申请更新索引,以保证系统不会降低其容错等级.

这些多副本在保障系统的可靠性的同时还可以增强系统的并行性,在存储介质变得越来越便宜的工业背景下,这种更加注重于系统响应及时性的工业需求就显得很理所当然.典型的情况如在上层应用请求数据传输时文件系统可以根据网络拓扑结构和负载情况来决定由哪个副本对上层应用做应答,这种情况特别适应它们的上层应用 MapReduce. 相对于基于纠删码的容错机制,这种方式无疑增加了存储开销.但是在响应客户应用的延迟要低很多,因为基于副本的容错机制不需要进行解码操作,相应的,因为不需要编码操作,向存储系统写入数据的过程也明显的效率要高很多.

基于多副本和纠删码是当今最主要的两种冗余容错策略,它们各有各的应用场景.简单的总结一下就是多副本机制在应对于要求响应的及时性的应用更为合适,纠删码能够在达到相同容错等级的情况下只需更少的额外存储开销^[2].

2.2 基于网络编码的容错机制

为了充分利用网络编码的优越性,该系统对每个原始文件进行切片并经过编码.所以,当存储有某一文件块的节点失效导致该文件容错等级降低时,系统会首先选取另外新的一个数据节点,用于存放新的文件块.该新节点会读取名字节点的编码矩阵和完好的文件块位置,根据编码矩阵选取合适的完好的编码文件块,通过相应的编码操作获得该失效节点上所存储的文件块并存储^[3-4].

典型的蝴蝶型网络编码如图 1 所示.

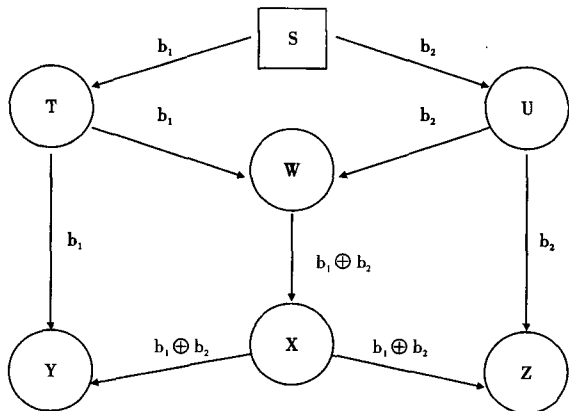


图 1 有限域为 $\{0,1\}$ 的典型的蝴蝶型网络编码

3 Cumulus 文件系统的设计

3.1 Cumulus 体系结构设计

Cumulus 采用 Java 语言开发,并在 GNU/Linux 操作系统下部署.一个典型的部署场景是,一台 GNU/Linux 操作系统上运行一个名字节点实例,作为 Master 中心服务器.而集群中的其它十几个到上千个 GNU/Linux 操作系统分别运行一个数据节点实例,作为 Slave 服务器集群.部署了该文件系统的任意一台机器可以充当 Client 的角色,发起文件存储申请以及文件读取申请.根据以上的需求分析,Cumulus 的体系设计如图 2 所示.

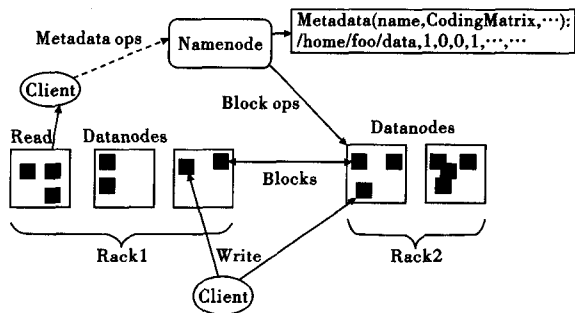


图 2 系统体系结构设计

系统中有唯一的名字节点作为 Master 服务器,主要有三部分功能:

- (1) 处理来自客户端的文件访问;
- (2) 管理文件系统的命名空间操作,如创建、打开、关闭、删除、重命名等操作;
- (3) 负责数据块到数据节点之间的映射.

数据节点作为 Slave 服务器,系统中有多个数据节点,数目大概在十几个到几百个的数量级,数据节点同样有三部分功能:

- (1) 管理挂载在节点上的存储设备;

(2) 响应客户端的读写请求;

(3) 从内部看,每个文件被分成一个或多个数据块,被存放到一组数据节点,在名字节点的统一调度下进行数据块的创建、删除和复制等实际操作。

Client 节点作为与用户的交互接口,可以发起存储申请、文件读取操作,并可以按照名字节点的要求将申请存储的文件按照指定的文件块大小(BlockSize)进行切片,以满足编码的需求^[4]。

3.2 基于网络编码的容错处理流程

对于 Cumulus 分布式文件系统,作以下建模。编码文件块向量 u 与原始文件块向量 c 有着以下的编码关系:

$u=cG$,这里的 $c=[c_1, c_2, c_3, \dots, c_k]$ 是原始文件, c_1 等是一个原始文件切片, $u=[u_1, u_2, u_3, \dots, u_n]$ 是编码过的切片集合,选用 MDS 编码方案,从 n 个编码过的切片中任选 k 个切片就可以完整的恢复出来原始文件。 G 为一个 $n \times k$ 的矩阵。

1001

比如,有编码矩阵 $G=1111$,这里 $n=4, k=3$,

0101

原始文件 $c=[c_1, c_2, c_3]$,那么有编码后的文件块有 $u_1=c_1+c_2$, $u_2=c_2+c_3$, $u_3=c_2$, $u_4=c_1+c_2+c_3$ 。

根据系统的网络拓扑结构以及系统中每个存储现有的负载情况,选定 n 个节点进行编码后的文件块的存储。

在存储有某一数据块的存储节点失效的情况下,系统启动修复进程。文中选取 u_1 所在的数据节点作为失效节点来作为典型场景进行分析,选取跟失效节点在同一个网段的节点作为 New Comer,如图 3 所示为修复流程。

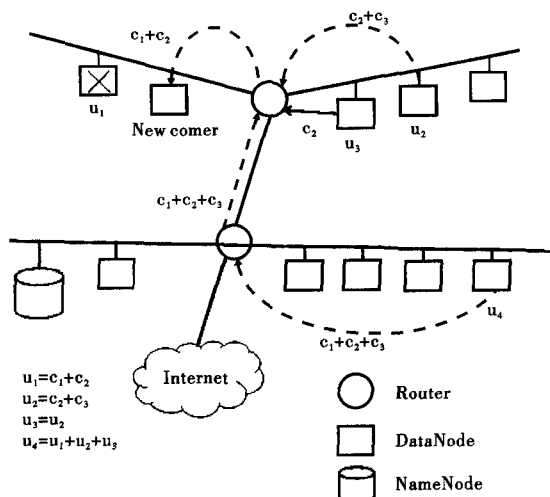


图 3 存储实例以及节点修复实例

由于 $u_2=c_2+c_3$, $u_3=c_2$, 并且它们在同一个网段,所以 Namenode 指导存储有这两个数据块的 Datanode 将 u_2 和 u_3 发送到连接这一网段的路由器,并做网络编码,获得文件块 c_3 ,然后将 u_4 发送到这一路由器,做编码操作,获得文件块 c_1+c_2 ,这一文件块正是所需要的,然后由路由器发送至 New Comer, New Comer 接收并存储该文件节点,然后发送确认信息, Namenode 在索引空间中进行记录。

4 Cumulus 文件系统的实现

4.1 读写流程

4.1.1 文件写流程

(1) 客户端申请写操作:Client 端向 Namenode 发起 RPC 远程调用,申请文件写操作请求;

(2) 名字节点创建空索引:Namenode 根据存储需求生成相应的编码矩阵(codingMatrix),计算文件切片所需要的切片大小(BlockSize),选择相应个数的 chosenDataNode 列表作为文件的存储节点,在索引空间创建新的文件,此时该索引文件没有 Block 列表与之对应;

(3) 客户端写数据:Client 开始写入文件,将文件切分成多个 packets,并在内部以 data queue 的形式管理这些 packets,并向 Namenode 申请新的 Block. 每个 Block 根据 codingMatrix 以及 chosenDataNode 列表计算该 Block 应该被发往存储节点数组 target[];

(4) 数据在数据节点的流动:以 pipeline(管道)的形式将 packet 写入 target 数组所包含的 DataNode 中. Client 端把 packet 以流的方式写入第一个 DataNode,该 DataNode 把该 packet 存储之后,再将其传递给在此 pipeline 中的下一个 DataNode,直到最后一个 Datanode;

(5) 确认流水线:最后一个 datanode 成功存储之后会返回一个 ack packet,在 pipeline 里传递至客户端,在客户端的开发库内部维护着 ack queue,成功收到 datanode 返回的 ack packet 后会从 ack queue 移除相应的 packet;

(6) 数据节点的编码:DataNode 内部运行一个接收 Block 线程(BlockReceiver),该线程在接收到每个 Blocks 之后进行相应的头部信息提取,进行相应的操作,并最终将编码完毕的 Block 放进 Block-Received 队列中,等待上传到 Namenode 进行索引维护;

(7) 释放资源:Client 端以这个 Block 的标示符

为信息,向 NameNode 发送 complete 消息,NameNode 接收到这个信息之后做一些资源释放等操作,比如释放租约等。

4.1.2 文件读流程

(1) 客户端申请读操作:Client 端向远程的 NameNode 发起 RPC 文件读取请求;

(2) 名字节点返回编码矩阵和文件块位置:NameNode 返回文件编码矩阵 codingMatrix 以及文件的全部编码文件块列表,对于每个编码文件块,NameNode 都会返回有该 Block 存储的 DataNode 地址;

(3) 客户端读取文件块:Client 端根据需求与文件块列表的前几个 DataNode 进行数据交互,来读取编码文件块,读取完当前 Block 的数据后,关闭与当前的 DataNode 连接,并为读取下一个编码文件块做 socket 连接;

(4) checksum 校验:读取完一个 Block 都会进行 checksum 验证,如果读取文件块时出现错误,客户端会抛出异常,放弃该 Block 并再多读一个 Block;

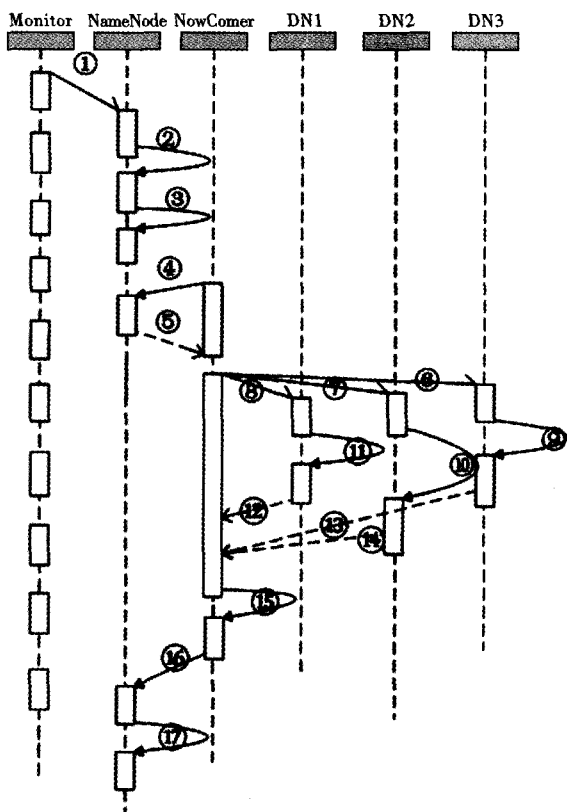
(5) 客户端解码:客户端根据读出的编码文件块以及解码矩阵进行数据解码,得到原始文件。

4.2 修复流程

每个 Datanode 节点周期性地向 Namenode 发送心跳信号.网络割裂可能导致一部分 Datanode 跟 Namenode 失去联系. Namenode 通过心跳信号的缺失来检测这一情况,并将 Datanode 标记为宕机,不会再将新的 I/O 请求发给它们.任何存储在宕机 Datanode 上的数据将不再有效. Datanode 的宕机会引起一些编码数据块丢失, Namenode 不断地检测这些需要复制的数据块,一旦发现就启动修复操作.有以下情况时,可能需要数据修复:某个 Datanode 节点失效,某个编码文件块遭到损坏, Datanode 上的硬盘错误.修复流程如图 4 所示。

(1) 检测文件块丢失:运行在 Namenode 上的容错进程检测到某个编码文件块丢失的时候就会将这个丢失的 Block 根据优先级放入 codedBlock-Missing 队列中,修复进程周期性的检测这个队列,队列非空的情况下唤醒进程,对于队列头部的 MissingBlock 调用 allocateNewComer 函数根据网络拓扑结构和负载情况分配新的存储节点;

(2) 计算修复流程: Namenode 通过网络拓扑结构和该文件现有的完好 Block 计算每个 Block 的传



①监视线程发现数据块丢失,通知 NameNode 启动修复;②为修复数据块分配 newComer;③在 newComer 修复任务队列中添加任务;④newComer 心跳;⑤修复命令;⑥、⑦、⑧读数据块;⑨、⑩、⑪初始化 blockSender;⑫、⑬、⑭发送数据块;⑮编码数据块;⑯上报新数据块信息;⑰更新 INodeFile

图 4 修复流程

输过程,即该 Block 传输到哪个路由器或者是 New Comer 节点以及在哪里期待与哪个 Block 做什么样的编码操作;

(3) 任务下放至数据节点: Namenode 将相应的操作命令放在 New Comer 节点中的 addToRecoverList 中,等待 New Comer 执行。

(4) 获取数据块: New Comer 根据队列的信息对于每个需要参与修复的 Datanode 借助心跳包发送 datanodeCommands, 这些 Datanode 收到 command 之后根据内容发送 Block,并在头部标示编码信息;

(5) 编码存储: Router 或者 New Comer 每接收一个 Block 就会根据头部标示信息进行相应操作. Router 根据头部信息确定该 Block 接收后编码缓存、接收后编码还是接收转发, New Comer 根据头部信息确定该 Block 接收后缓存、接收后编码还是

编码后写磁盘,并放入 BlockReceived 列表中,等待向 Namenode 写索引^[6].

5 系统性能分析

由于面向网络编码的分布式文件系统采取了基于纠删码的容错机制,使得文件在写流程中增加了编码操作,在读文件流程中增加了解码操作,所以,该系统的响应时间、读写速度等指标会稍微落后于 HDFS.但是,在为了达到相同等级的容错级别的过程中,采取的纠删码的文件系统的额外存储开销要远低于基于副本的额外开销^[2].

由于节点失效的修复过程应用了网络编码,因此,网络带宽的利用率得到了很大的提升.针对图 2 中的情况,在 New Comer 所在的网段,传统的修复模式需要传输三个 Block 大小的数据量,而在应用了网络编码,在该网段只需要传输一个 Block 的数据量,极大的节省了网络带宽的消耗量.

6 结束语

网络编码是一种比较新的研究课题,它很多方面的优越性都得到了很好的体现.为了研究网络编码在分布式文件系统中的应用以及它所能带来的优越性,文中实现了一个面向网络编码的分布式文件系统.该系统采用纠删码的冗余容错机制,能够很自然的引入网络编码,利用该系统可以做很多理论上

的研究与应用.

参考文献:

- [1] 陶少国,黄佳庆,杨宗凯,等. 网络编码研究综述[J]. 小型微型计算机系统, 2008,29(4): 583-592.
- [2] Weatherspoon H, Kubiatowicz J. Erasure coding vs. replication: A quantitative comparison[C]// Proceedings of the 1st International Workshop on Peer-to-Peer Systems. USA: Cambridge, 2002: 328-337.
- [3] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, et al. The hadoop distributed file system[C]// Proceedings of MSST2010. USA: Nevada, 2010.
- [4] Alexandros G Dimakis, Kannan Ramchandran, Yunnan Wu, et al. A survey on network codes for distributed storage[J]. Proceedings of the IEEE, 2011,99(3):476-489.
- [5] HDF Architecture Guide(OL)[EB/OL]. http://hadoop.apache.org/common/docs/current/hdfs_design.html.
- [6] Yeung R W, Li S Y R, Cai N, et al. Network coding Theory[J]. Foundations and Trends in Comm and Info Theory, 2005,2(5): 241-381.

作者简介:

陈运海 男,硕士研究生.研究方向为分布式文件系统与网络编码.

陈正亮 男.研究方向为分布式文件系统与网络编码.

王钦辉 男,博士研究生.研究方向为认知无线网络.

(上接第 73 页)

4 结束语

为了确保电子政务系统信息安全,周密的电子政务系统信息安全风险分析,是制定准确可靠、切实有效的安全防护措施的必要前提.传统的电子政务系统信息安全评估方法的不足,文中提出了基于危险理论的电子政务系统信息安全风险评估方法.结果表明,该方法具有较高的评估精度,评估方法合理可行,具有一定的实践推广价值.

参考文献:

- [1] 冯惠玲,王健.电子政务建设中的文件管理风险探析[J]. 中国行政管理,2005(4):62-66.
- [2] 汤志伟,杜人杰,高天鹏.基于 ISM 模型的电子政务信息系统风险分析[J]. 电子科技大学学报,2005,34(2): 251-253.

- [3] 蔡昱,张玉清,冯登国.风险评估在电子政务系统中的应用[J]. 计算机工程与应用,2004,26(4):155-159.
- [4] 陈涛,冯平,朱多刚.基于威胁分析的电子政务信息安全风险评估模型研究[J]. 情报杂志,2011,30(8):94-99.
- [5] 刘越男.识别电子政务系统中的文件风险[J]. 档案学通讯,2006(2):52-55.
- [6] 吴叶科,宋如顺,陈波.基于梯形模糊 AHP 的信息安全风险综合评估[J]. Journal 6, 2011,47(34):111-113.
- [7] 郑雷雷,宋丽华,郭锐,等.故障树分析法在信息安全风险评估中的应用[J]. 计算机科学,2011,38(10A): 106-109.
- [8] 申时凯,余玉梅.模糊神经网络在信息安全风险评估中的应用[J]. 计算机仿真,2011,28(10):91-94.

作者简介:

张本群 女,(1975-),硕士研究生,讲师.研究方向为计算机算法分析.