

# 分布式存储方案的设计与研究

胡文波,徐造林

(东南大学 计算机科学与工程学院,江苏 南京 211189)

**摘要:**针对基于纠删码的分布式存储方法中信息分割算法 IDA 在运算中涉及构造拆分矩阵,计算开销大,且单纯基于纠删码的方案无法确保所存储数据的完整性、机密性等特性,只能容忍系统中存在的良性故障,无法容忍入侵者的恶意入侵。提出了分布式系统中一种基于 Tornado 码的浏览器-服务器工作模式的数据存储方案。在数据写入过程中通过构造编码后数据分块的 Hash 值级联(即数字指纹),并与每个数据分块一起分布在存储服务器集中的不同服务器中,当需要读出时对分块及数据指纹进行验证,然后利用 Tornado 译码方法恢复原始文件,即可实现 Byzantine 环境数据的完整性保护,并提高了系统的容错能力。

**关键词:**存储系统;拜占庭错误;冗余;Tornado 码

**中图分类号:**TP301

**文献标识码:**A

**文章编号:**1673-629X(2010)04-0065-04

## Design and Research on Distributed Storage Scheme

HU Wen-bo, XU Zao-lin

(School of Computer Science & Engineering, Southeast University, Nanjing 211189, China)

**Abstract:** In the distributed data storage scheme, in order to solve the problem of large calculating costs brought by the construction and split of matrix involved in the IDA algorithm, and a simple program based on erasure codes can not ensure the data integrity, confidentiality and other features. Presents a kind of browser-server mode of data storage scheme based on Tornado code in the distributed system. In the process of data writing, through the construction of the Hash value class of data piece after coding, namely the numerical fingerprint, along with each data block stored in different servers which can be veriflicated when it is needed by the Tornado decoding method, thus the integrity of the Byzantine environment data can be fully protected and improve the system's fault tolerance.

**Key words:** storage system; Byzantine fault; redundancy; Tornado code

## 0 引言

Internet 和网络技术的飞速发展极大地推动了分布式存储技术的进步,同时也给分布式存储技术不断提出新的需求。目前,分布式存储技术的发展趋势和主要的研究热点<sup>[1,2]</sup>如下:

(1)高性能。对分布式存储系统的每一个用户,系统都应该能够提供始终如一的高性能存储服务。不考虑硬件和网络设施的因素,系统应该尽可能地克服或缓解网络环境的动态性和不可预知性对服务性能造成的影响。

(2)高可靠性。分布式环境通常都有高可靠性的需求,用户将文件保存到分布式存储系统的基本要求是数据可靠。系统应该采用有效的容错机制,使得一

些常见故障<sup>[3]</sup>(如节点离线或失效、网络断开等)对用户透明,用户访问文件时,文件不会因为网络故障或部分节点不在线而不可得,使用户在动态变化的网络环境下获得高可靠的文件服务。

(3)高可扩展性。分布式存储系统要能适应节点规模和数据规模的增长必须具有高可扩展性,系统的存储容量可以随着用户存储需求的增长而增长,以支持海量存储。

(4)透明性。如果一个分布式存储系统让用户和应用程序感觉和本地存储空间一样,就说它具有透明性。分布式存储系统通过内部实现机制和用户接口为用户提供透明的存储服务。

## 1 系统总体框架

文中将基于 Tornado 码,给出一种分布式系统中实用的容忍入侵数据存储服务方案,能够在 Byzantine 入侵者存在的情况下维持系统的可靠性和可用性。系统的框架结构如图 1 所示。

收稿日期:2009-08-08;修回日期:2009-11-10

作者简介:胡文波(1985-),男,安徽巢湖人,硕士研究生,研究方向为计算机体系结构、嵌入式系统;徐造林,副教授,研究方向为计算机体系结构、嵌入式系统、控制系统及其应用。

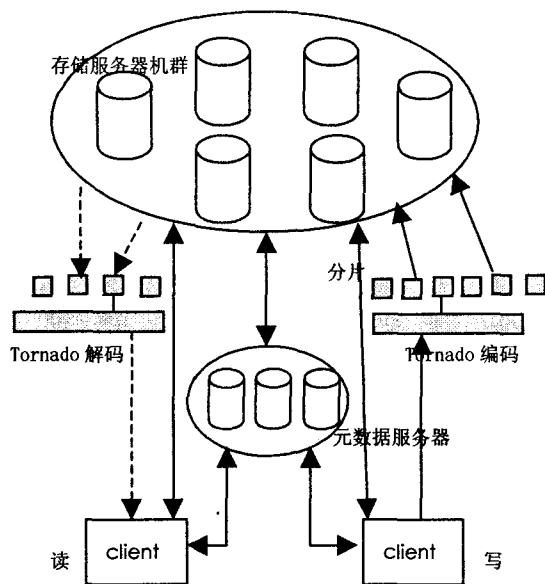


图1 系统总体框架

系统采用 B/S 的工作方式, 客户和服务端间存在认证的加密通道(可由 SSL 实现)。存储服务器的数据库中, 所存储的每个数据条目都包含有 4 个域: 文件标识符 FID、版本信息、数据分片内容和所有数据分片的指纹。初始化时, 所有数据项的版本信息均为 0; 当针对某 FID 执行一次写操作时, 新写入的数据项版本信息为该 FID 所对应的所有数据项中最高版本信息加 1; 针对某 FID 执行一次读操作时, 存储服务器将该 FID 对应的所有数据项中最高版本的数据项中的数据内容返回给客户。

当客户需要对一个文件进行访问时, 它首先向目录服务器提交请求, 目录服务器根据客户需求定位到用户所需访问的文件标识符, 客户使用该文件标识符就可向存储服务器集请求相应操作。存储服务器集中的服务器可以在一个局域网内分布, 也可以跨网络域分布在整个 Internet 中, 其主要思想就是确保服务器集中的部分服务器的被攻陷不会影响整个服务器集的数据存储功能, 即便是元数据服务器被攻陷。在存储服务器集中, 文件使用安全 Tornado 码的方式被编码, 编码后的数据分片被分布存储在服务器集的各个服务器中。由于安全 Tornado 码具有纠错和查错的功能, 当服务器集中的某些服务器崩溃或被攻击者成功控制时, 其余服务器仍能够恢复出原始信息。

具体的读写过程如下:

假设一个客户要写某个文件 F, 首先向目录服务器发出请求, 目录服务器记录该文件的相应信息, 包括文件所有者、所在组以及存取模式和许可信息等, 产生其存储结点并向各存储结点服务器 S 发送该文件相关信息, 同时产生一个并向客户返回一个文件标识符

FID(该 FID 将作为存储服务器中存储文件 F 编码后数据分块时的第一索引)和分配的服务器结点  $S_1 - ID, S_2 - ID, \dots, S_m - ID$ ; 客户根据返回的 FID, 向分配的存储服务器集中的服务器发出以该 FID 为标志的写句柄请求; 存储服务器在收到客户请求之后首先验证服务器自身状态和用户的身份及其权限, 如权限允许, 则查询以该 FID 为标志的所有已经存储的数据分块的版本信息, 以检索到的最高版本加 1 作为当前写操作的版本信息, 生成一个写句柄返回给客户, 该写句柄中包含有文件标识符 FID 以及该 FID 加 1 后的版本信息等, 可视做客户对存储服务器集执行写操作时的许可证; 收到来自服务器集的句柄之后, 客户先将文件 F 分割为  $m$  个大小为  $F/m$  的分段, 然后使用安全 Tornado 编码方法对这  $m$  个文件分段进行编码, 生成  $n$  个数据分块  $D_1, D_2, \dots, D_n$ , 以及关于这  $n$  个数据分块的一个数字指纹 CC。随后, 客户就可以使用句柄向服务器集的服务器结点  $S_i$  发出对数据分块  $D_i$  和 CC 的写请求; 存储服务器集中的每个服务器在收到客户写请求之后验证句柄的有效性, 如有效, 则以文件标识符 FID 为第一索引, 以句柄中包含的版本信息为第二索引将该分片写入其数据库中, 并且成功存储的服务器结点返回信息给目录服务器。

类似地, 当客户需要读文件 F 时, 首先向目录服务器发送请求, 目录服务器根据所存储的关于文件 F 的路径以及包括文件所有者、存取模式和许可模式等信息判断是否向用户授权。如授权, 则生成并向客户返回文件 F 的存储结点信息, 客户向服务器发出读句柄请求, 存储服务器在收到客户请求之后首先验证用户的权限, 如权限许可, 则查询以 FID 为第一索引的所有数据项中的版本信息, 以检索到的最高版本作为当前读的版本信息, 并生成一个读句柄返回给客户; 客户使用该句柄向存储服务器发出读请求; 存储服务器在收到客户读请求之后验证句柄的有效性, 如有效, 则将由 FID 及版本信息作为复合索引的数据项返回给客户, 其中包括一个数据分块和一个数字指纹; 如果存在来自  $n - t$  个存储服务器的数据项满足以下两条:

(1) 这  $n - t$  个数字指纹都是相同的;

(2) 每个数据项中的数据分块的 Hash 值都与其数字指纹匹配, 则利用 Tornado 码译码方法<sup>[4]</sup>, 就可从这  $n - t$  个数据块中恢复出原始文件。

## 2 安全 Tornado 码存储方案

### 2.1 Tornado 码简介

Tornado 码是一种建立在非规则图上的低密度校验码。其编码算法主要包括以下步骤:

(1)数据分割:将数据  $D$  分割为  $m$  个大小为  $b$  的数据分组,即  $D = [d_1, d_2, \dots, d_m]$ ;

(2)编码:对  $m$  个数据分组进行编码,产生出  $n$  个数据分组(这里,  $e_r = m/n$  称为编码率);

(3)译码:从编码后的  $n$  个数据分组中获取  $r$  个数据分组( $r > m$ ),对它们进行译码得到原来的  $m$  个数据分组;

(4)数据恢复:由译码得到的  $m$  个数据分组恢复数据  $D$ 。

Tornado 码是系统码,所以编码后的前  $m$  个数据分组就是原来的  $m$  个数据分组,这  $m$  个数据分组称作信息块,其余的  $n - m$  个数据分组称作校验块。因此,编码后的结果包括  $d_1, d_2, \dots, d_m$  这  $m$  个原始信息块和从  $D_1$  到  $D_{\beta m}$  这  $\beta m$  个校验块,即  $n = (1 + \beta)m$ 。

利用二分图  $B$  对数据块  $d_1, d_2, \dots, d_m$  进行编码,产生包括  $d_1, d_2, \dots, d_m$  在内的  $(1 + \beta)m$  个数据块,即产生  $m$  个信息块和  $\beta m$  个校验块。二分图  $B$  定义了信息块与校验块之间的映射关系,每个校验块可以由其相邻的信息块进行异或运算得到(见图 2(a))。

如果与某一校验块相邻的所有信息块中存在一个丢失的信息块,那么可以通过该校验块和其它信息块的异或运算得到丢失的信息块(见图 2(b)),这就是 Tornado 码译码的基本思想。

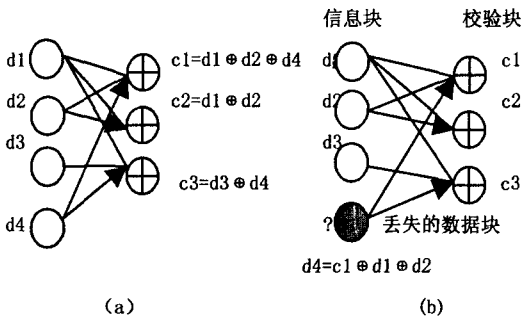


图2 Tornado 码译码过程

利用级联二分图对数据分组实现编码,这样即便某些校验块丢失也不影响原数据恢复。级联二分图(见图 3)由若干个二分  $B_1, B_2, \dots, B_k$  连接而成,其中对于每一级非规则二分图来说,相邻的不规则二分图  $B_i$  的输出为后面一级二分图  $B_{i+1}$  的输入。构造图中的所有节点输出的分组构成了编码后的所有分组的级联二分图。利用  $B_1$  可以由  $m$  个信息块产生  $\beta m$  ( $\beta < 1$ ) 个校验块,这  $\beta m$  个校验块就是  $B_2$  的信息块,利用  $B_2$  又可以产生  $\beta^2 m$  个校验块,依此类推,利用  $B_i$  可以由  $\beta^{i-1} m$  个信息块产生  $\beta^i m$  个校验块。因此,利用编码  $C(B_1, B_2, \dots, B_k)$  产生校验块的数目为  $\sum_{i=1}^k \beta^i m$ 。

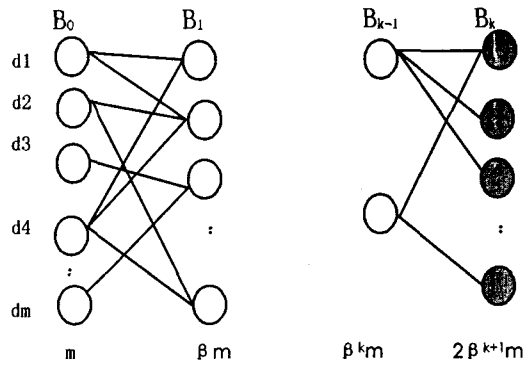


图3 级联二分图

通常,解码从最高级二分图开始,首先利用  $B_{k-1}$  和  $B_k$  得到它们的信息块,即  $B_{k-2}$  的校验块;然后,依次得到  $B_{k-3}, \dots, B_1, B_0$  的校验块;最后,得到  $B_0$  的  $m$  个信息块。当然,如果在解码开始时就已经得到了  $B_i$  ( $0 \leq i < k$ ) 的所有校验块,那么,可以直接从  $B_i$  开始解码。

## 2.2 数据分片分发策略

在该网络存储系统中,数据是经过分片的方式存放在网络中的多个存储结点上的。如何选择有效的数据放置策略来使系统具有稳定的可靠性和良好的性能至关重要。通常数据放置策略分为顺序放置策略和随机放置策略<sup>[5]</sup>。前者当发生故障的结点数量较多时,恢复系统可靠性的开销比较大而后者保证数据均匀的分布在系统中,从整体上看有利于存储的负载均衡,但是数据访问的本地性较弱,对系统的性能影响较大,当结点发生故障时恢复所丢失数据的开销远小于前者,然而当系统随机的出现较多的结点故障时,故障范围覆盖各副本放置目标的概率会比较大,因此随机放置策略的容错能力相对较差。

该系统的设计目标允许存储节点动态的进出网络,所以数据的放置策略必须能够适应网络中节点数量的变化,另外为了保证系统的可靠性还必须设定合适的冗余机制。系统中采用了 Gossip<sup>[6]</sup>传播更新算法的存储转发随机放置策略。Gossip 算法过程主要为:①更新操作的发起者将更新消息发给所有节点中  $m\%$  的接受者,并将每个消息设置计数器  $n$ 。②接受者收到一个消息后,如果  $n \neq 0$ ,则发送给另一个接受者,并将其计数器减 1。结合 Gossip 算法,该校系统采用了随机选择的发送方式,通过存储转发来生成数据块的多个冗余副本,能够较好地保证系统可靠性问题。

## 2.3 元数据安全

虽然相对于整个系统的存储量,元数据的存储量是较小的,但是元数据的访问量占整个系统的访问量的 50%~80%,因此对元数据的安全是一个潜在的瓶

颈,提供高可靠的元数据访问服务,对存储的元数据进行冗余处理才能保证整个存储系统的性能。副本的冗余方式可分为两种:对要存储的元数据保持多个完整的副本;将要保存的元数据分成碎片,对每个碎片保存多个完整的副本,再将这些副本分发到其他元数据服务器节点上。当出现节点失效时,通过访问存储这些副本的节点实现对数据的冗余。当这些存储副本的节点发生 Byzantine 错误时,通过使用 Byzantine 将军算法,由 Lamport 等人<sup>[7]</sup>提出,他们证明了在系统中,对于  $f$  个 Byzantine 错误节点,需  $3f + 1$  个或以上副本的冗余才能使系统冗余 Byzantine 错误。在元数据副本数据不一致时能实现对该错误的冗余,保持系统可靠性、元数据一致性和系统安全性。

### 3 性能分析

#### 3.1 可靠性

在高动态的分布式存储系统中,结点在任意时刻可能表现出随意性的错误,致使数据的不一致,即 Byzantine 错误,要保证系统正常运行,就需使用 Byzantine 容错技术。目前较多使用的是完全副本冗余和纠删码冗余。

2002 年, OceanStore 项目的研究者 Weatherspoon 等人<sup>[8]</sup>就用量化的方法分析了纠删码和副本方式冗余对系统可靠性的影响。在分布式系统中使用纠删码冗余,可以在与副本冗余得到相同可靠性的条件下,极大地节约系统中的存储空间和维护带宽;反之,若使用相同的存储空间和维护带宽,纠删码方式能够极大地提高系统的可靠性。因此,纠删码有利于提高系统的可靠性。但在纠删码方式下,要修复一个丢失的冗余碎片就需要有一个完整副本,严重浪费系统带宽。所以采用纠错码与副本方式结合的两层冗余方法:首先对存储数据作纠错码冗余,然后对每个冗余碎片作副本冗余,这样丢失一个碎片就只需从另一个碎片的副本处读取同样大小的数据修复即可。该方法可提高此划分的带宽利用率并有效降低通信时延,从而在一定程度上提升整个系统的性能。

#### 3.2 数据完整性

对文件进行分片存储,用户访问数据时需要进行重新组合,这时需要保证每个分片的完整性,安全 Tornado 的基本思想就是计算 Tornado 编码之后的每一个数据分片的 Hash 值,将编码后的  $n$  个数据分块的 Hash 值进行级联,将级联后的结果称作数字指纹,并与每一个数据分块一起分布在存储服务器中;需要解码时,首先对来自存储服务器集中  $r$  个不同服务器所存储的数据分块及数字指纹进行验证,如果这  $r$  个数

字指纹都是相同,且与每个数据分块的 Hash 值匹配,则通过验证,然后利用 Tornado 码译码方法从这  $r$  个数据分块中恢复出原始文件即可。当攻击者能够成功攻陷的服务器个数不超过  $(n - r)$  时,以上方案能够实现 Byzantine 环境中对基于 Tornado 码的存储方案的完整性保护。

#### 3.3 数据一致性

系统可以由多个节点并发访问文件数据。只要获取相应的文件访问权限,多个用户可以同时访问同一文件数据。对文件的访问由元数据服务器管理,元数据服务器在执行某个操作前都要获得一系列锁,例如,它要对  $/d1/d2 \dots /dn/leaf$  执行操作,则它必须获得  $/d1, /d1/d2, \dots, /d1/d2 \dots /dn$  的读锁,  $/d1/d2 \dots /dn/leaf$  的读锁或写锁(其中 leaf 可以是文件也可以是目录)。对文件操作的并行性和数据的一致性就是通过这些锁来实现的。

### 4 结束语

文中提出了分布式系统中实用的容忍入侵数据存储方案,通过使用 Tornado 码作为基本的编码手段,能够实现在线性时间内的数据编译码,避免了现有方案计算开销过大等问题。对于数据分片,采用基于 Gossip 算法的数据放置策略,该策略在一定程度上避免了存储服务器结点造成的数据丢失,进一步提高了系统的容错能力。未来工作将着重于元数据服务器的性能优化和系统的具体实现。

#### 参考文献:

- [1] 徐非,杨广文,鞠大鹏.基于 Peer-to-Peer 的分布式存储系统设计[J].软件学报,2004,15(2):268-277.
- [2] 鲍捷,宋靖雁.分布式网络计算机域的一种系统模型及其文件系统[J].计算机应用与软件,2006,23(5):86-88.
- [3] 杨磊,黄浩,李仁发,等.P2P 存储系统拜占庭容错机制研究[J].计算机应用研究,2009,26(1):4-8.
- [4] 王意洁,卢锡城.基于 Tornado 码的复制算法[J].国防科技大学学报,2004,26(3):39-42.
- [5] 田敬,代亚非.P2P 持久存储研究[J].软件学报,2007,18(6):1379-1399.
- [6] Lin M, Marzullo K. Directional Gossip: Gossip in a Wide-Area Network[R]. San Diego: Dept of Computer Science and Eng, University of California, 1999.
- [7] Lamport L, Shostak R, Pease M. The Byzantine generals problem[J]. ACM TO PLAS, 1982, 4(3):382-401.
- [8] Weatherspoon H, Kubiatowicz J. Erasure coding vs. replication: A quantitative comparison[C]//In: Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems. Berlin: Springer, 2002:328-337.