文章编号:1001 - 9081(2006)05 - 1102 - 04

一种基于信息分散算法的分布式数据存储方案

屈志毅,苏文洲,赵 玲
(兰州大学 信息科学与工程学院,甘肃 兰州 730000)
(shixp05@st.lzu.edu.cn)

摘 要:针对分布式数据存储方案中,基于复制的方法和基于秘密共享的方法存在存储开销过大的问题,提出了分布式系统中一种基于 IDA 码的客户-服务器工作模式的数据存储方案。该方案在数据写入过程中通过构造编码后数据分块的 Hash 值级连,即所谓的数字指纹,可实现 Byzantine 环境数据的完整性保护。

关键词:分布式数据存储;信息分割算法;Hash 值级连
中图分类号: TP309.2 **文献标识码**:A

Distributed data storage method based on information decentralization algorithm

QU Zhi-yi, SU Wen-zhou, ZHAO Ling
(*School of Information Science & Engineering, Lanzhou University, Lanzhou Gansu* 730000, *China*)

Abstract: In the distributed data saving project, in order to solve the problem of large saving expense aroused by replication method and secret sharing method, a data saving project in the distributed system based on the customer - server mode of IDA code was put forward. In the process of data writing, through the construction of the Hash value class of data piece after coding, namely the numerical fingerprint, the integrality of the Byzantine environment data can be fully protected.

Key words: distributed data storage; information division algorithm; Hash class connect

0 引言

在当前基于冗余的分布式存储系统设计中,主要设计思路可以分为两类:一类是基于复制的方法,另一类是基于秘密共享的方法。在基于复制的分布式存储系统方面,MIT 的 Castro 等人在这方面进行了深入研究,他们采用复制的状态机方法[1]构建了一种能够容忍 Byzantine 故障的文件存储系统[2]。同样使用复制的状态机方法,MIT 的 Rodrigo Rodrigues 等人则首次设计并实现了一种能够实现动态成员变化的容忍 Byzantine 故障的文件存储系统[3]。

Quorum 方法[4]也是实现基于复制的分布式存储的一种常用方法。早期基于 Quorum 方法的系统主要考虑的是处理良性失效(benign failures)的问题[5];后来,人们又研究了基于 Quorum 方法的 Byzantine 失效问题[6]。例如 Phalanx 和 Fleet[7]等,都是基于这些研究结果所开发的分布式存储系统。此外,文献[8]中给出了一种基于 quorum 的自适应 Byzantine 存储模型框架,其中的 Quorum 协议能够在系统出现故障时自动重配置。

总体上讲,基于复制的分布式存储方案所需的存储空间太大,不是存储优化的,为了保护 f 个可能出现 Byzantine 故障的服务器,对于每个数据对象至少需要维护 $3f+1$ 个复制品。

佐治亚理工学院(Georgia Institute of Technology)的 AgileStore 研究项目[9]结合使用了秘密共享方法、复制方法和 Quorum 方法,设计、实现并评估了一个集成的敏捷存储架构,其目的是想在安全性、容忍入侵特性和性能之间达成一种折中。与上述方法类似,文献[10]中也结合完善多份额秘密共享和复制技术,设计了称之为 GridSharing 框架的容忍入侵存储结构。尽管上述研究都声称其方案的计算开销较低,存取速度较高,但我们认为这种秘密共享 + 复制的方式在存储空间方面的开销太大,不是理想的分布式存储方式。例如,采用 (n, t) 秘密共享方案的话,存储一个长度为 l 的文件,所需的存储空间至少为 $l*n$。

与基于复制或秘密共享的分布式存储方法相比,基于信息分散算法(Information Dispersal Algorithm, IDA)的方法需要的存储空间会大大减少(对于 (M, N) IDA 码而言,存储一个长度为 l 的文件,所需的存储空间为 $l \cdot \frac{N}{M}$)。

本文将基于 IDA 码,给出一种分布式系统中实用的容忍入侵数据存储方案,该方案使用密码学机制,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,并且具有很低的存储开销、计算开销和带宽开销。

1 IDA 码简介

IDA 算法中假设有一个大小 $|F|$ 的文件 F,该文件被分成 n 个文件分块,每个分块的大小为 $|F|/m$($n \geq m$,且通常情况下 $n/m \cong 1$,$n - m \geq k$,k 为文件受到攻击后,受损文件分块数目的上限,当大于这个上限后,文件将无法得到重组),从 n 个分块中任意取出 m 块就足够重建该文件,且分块文件

收稿日期:2005 - 11 - 11;**修订日期**:2006 - 03 - 02

作者简介:屈志毅(1957 -),男,陕西商县人,教授,主要研究方向:网络、多媒体信息处理、模式识别、图像处理; 苏文洲(1974 -),男,陕西眉县人,硕士研究生,主要研究方向:网络、多媒体信息处理; 赵玲(1978 -),女,陕西子洲县人,硕士研究生,主要研究方向:网络、多媒体信息处理.

的长度总和为$(m/n)\cdot L$。

下面是对该算法的实现进行详细描述:

首先,假设文件F由字符串组成,表示为$F=b_1,b_2,\cdots,b_N$,其中字符b_i可以认为是取值范围是$[0,\cdots,B]$($B=2^l-1$,l为一个字节拥有的位数)的整数;我们取最小的大于B的素数p作为该文件中字符运算的模,即该字符串中所有的元素都是以z^p为上限。

其次,选择一个合适的整数m,使$n=m+k$满足$m/n<1+\varepsilon,\varepsilon>0$。

然后,选择n个向量$a_i=(a_{i_1},a_{i_2},\cdots,a_{i_n})\in Z_p^m$,$1\leqslant i\leqslant n$,使该$n$个向量中任意不同的$m$个向量非线性相关。这样,随机从$(a_1,a_2,\cdots,a_n)$中抽取$m$个向量,这$m$个向量是非线性的可能性是非常大的。

最后,利用选择的向量组对矩阵F进行拆分与重组:

把文件F分成长度为m的序列,因此:

$F=(b_1,b_2,\cdots,b_m),(b_{m+1},b_{m+2},\cdots,b_{2m}),\cdots$

令　$s_1=(b_1,b_2,\cdots,b_m),s_2=(b_{m+1},b_{m+2},\cdots,b_{2m}),\cdots,$

则　$F_i=c_{i_1},c_{i_2},\cdots,c_{1_{N/m}}$,　　$i=1,\cdots,n$

其中,$c_{i_k}=a_i\cdot s_k=a_{i_1}\cdot b_{k-1_{m+1}}+\cdots+a_{i_m}\cdot b_{k_m}$

$|F_i|=|F|/m$

如果文件F的m个分块F_1, F_2,$\cdots$, F_m已经给出,那么可以依照以下步骤重建文件F。

让$A=(a_{ij})$,$1\leqslant i,j\leqslant m$是一个$m\times m$矩阵,其中第$i$行为$a_i$,可以看出:

$$A\cdot\begin{bmatrix}b_1\\ \vdots\\ b_m\end{bmatrix}=\begin{bmatrix}c_{11}\\ \vdots\\ c_{m1}\end{bmatrix}$$

因此,$$\begin{bmatrix}b_1\\ \vdots\\ b_m\end{bmatrix}=A^{-1}\cdot\begin{bmatrix}c_{11}\\ \vdots\\ c_{m1}\end{bmatrix}$$

其中$b_j=a_{i_1}c_{1_k}+\cdots+a_{i_m}c_{m_k}$,$1\leqslant j\leqslant N$。

2　安全 IDA 存储方案

由于 IDA 码是一种纠删码,不具有纠错功能,因此单纯使用 IDA 编码的存储方案仅能保证信息分片丢失时数据对象的安全性保护,这对应于存储服务器的崩溃情况。

为了能够容忍 Byzantine 入侵者对系统造成的破坏,基于对 IDA 码的完整性验证,我们给出了一种安全 IDA 方案,以防止分布式存储系统中的 Byzantine 攻击者的破坏。该方案中使用了一个单向 Hash 函数H,该函数能够保证资源受限的攻击者不能够在多项式时间内找到两个不相同的字符串x和y,满足$H(x)=H(y)$。

安全 IDA 的基本思路就是对于 IDA 编码之后的每一个数据分块,计算它的 Hash 值,以便在译码前验证数据分块的完整性,即实现分块的“查错”功能。

然而这样又带来了另外一个问题,就是 Hash 值本身也可能面临完整性攻击。例如对于数据分块x,尽管攻击者不可能在多项式时间内找到一个满足$H(x)=H(y)$的数据分块y,这里$x\neq y$,但它如果用另外一组不同的数据$(z,H(z))$代替$(x,H(x))$,则系统无法对此进行正确验证。

为了解决这个问题,在安全 IDA 方案中,将编码后的n个数据分块的 Hash 值进行级连,将级连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集中的不同服务器中;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 IDA 码译码方法从这r个数据分块中恢复出原始文件即可。

显然,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 IDA 码的存储方案的完整性保护。

假定 IDA 码的编/译码算法分别为 ENCODE( )和 DECODE( ),且编码之后的数据分块数目为n,译码时至少需要r个正确的数据分块(这里$r>n/2$),以对数据D的操作为例,假设由n个服务器$S_1,S_2,\cdots,S_n$组成分布式的存储服务器集,且攻击者能够成功攻陷的服务器个数不会超过$(n-r)$,则基于安全 IDA 码的数据存储方案的算法细节可描述如下:

需要实施文件存储时:

客户端

(F1, F2, ..., Fn) := ENCODE(F)
CC := MAKE_DISTRIBUTED_FIGURES(F1, F2, ..., Fn);
for all Si ∈ {S1, ..., Sn} do
  SEND (DID, Write-Request, CC, Fi) to Si;
  // "DID"表示 F 的标识符
end for
MAKE_DISTRIBUTED_FIGURES(F1, F2, ..., Fn):
for all i ∈ [1, n] do
  Hi := H(Fi)
  CC := H1 | H2 | ... | Hn　　// "|"表示对数据的级连
end for
RETURN (CC)

服务器端

RECEIVE_WRITE_REQUEST(DID, Write-Request, CC, Fi) from user C:
if the request is authorized and Hi = H(Fi) then
  Records := Records ∪ {(DID, CC, Fi)}
  OutPut (DID, out, stored)
end if

需要重构文件时:

客户端

READ ( ):
for all i ∈ [1, n] do
  SEND (Read-Request, DID) to Si
end for
Ω = ∅
repeat
wait for a message (block, DID, CC, Fi) from server Si such that Hi = H(Fi)
Ω = Ω ∪ {(CCi, Fi, Ki)}
until there exists a CC and a set S ⊆ [1, n]
such that (|S| = r) ∧ (∀j ∈ S: ∃CCj: CCj ∈ Ω ∧ CC = CCj)
F := DECODE(Fi: i ∈ S)
RETURN(F)

服务器端

RECEIVE_READ_REQUEST (read-Request, DID) from user C:
CC := Records(DID).Dfingerprint
Fi := Records(DID).Dcontent
SEND (block, DID, CC, Fi) to user C

定理 1　当$n\geqslant 2t+1$且$t+1\geqslant r$时,如果 IDA 方案能够容忍不超过$(n-r)$个丢失的信息份额,则当系统中至多有t个服务器被具有 Byzantine 行为的攻击者成功攻陷时,上述安全 IDA 方案能够以很高的概率允许客户重构文件。

证明 当$n \geq 2t+1$且Byzantine行为的攻击者成功攻陷的服务器个数不超过t时：

1）由$n-t \geq 2t+1-t = t+1$可知，来自存储服务器集中$n-t$个不同服务器所存储的数字指纹中必然有一个指纹是正确的，由于我们的协议要求这$n-t$个指纹相同，这就保证了这$n-t$个服务器中所存储的数字指纹的正确性。

2）由于攻击者成功攻陷的服务器个数不超过t，因此保证了系统中$n-t$个正确的指纹的存在性。

对于这$n-t$个服务器中的任何一个服务器F_i，攻击者都无法使用另外一组不同的数据$(z, H(z))$代替它所存储的$(F_i, H(F_i))$。由于之前已经验证过这$n-t$个服务器中所存储的数据分块及数字指纹的一致性，因此保证了这些服务器中的任意一个所存储的数据分块的Hash值都与其数字指纹匹配。根据Hash函数的特性可知，这$t+1$个数据分块是正确的。

又由于IDA方案能够容忍不超过$(n-r)$个丢失的信息份额，因此当$n-t \geq t+1 \geq r$时，基于这$n-t$个数据分块值，就可以利用IDA码译码算法以很高概率恢复出原始文件。

定理2 当正确地实施了文件存储之后，如果对于文件F的两次不同读请求，返回的结果分别是F'和F''，则$F' = F''$。

证明 由于两次读操作均需要验证$n-t$个相同的数字指纹，由$2(n-t) \geq (n-t)+(2t+1-t) = n+1$可知，至少存在一个服务器$S_i$，其所存储的数字指纹在两次读操作中均被验证成功，因此两次读操作中使用的数字指纹必定相同，即文件F'和F''对应着同样的数字指纹值。由Hash函数的防碰撞性特性可知，$F' = F''$。

3 系统总体框架结构

系统的框架结构如图1所示。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。

系统采用B/S的工作方式，整个系统由4个协议组成：客户方的write()、客户方的read()、存储服务器端的守护程序和目录服务器端的守护程序。客户和服务器间存在认证的加密通道（可由SSL实现）。数据存储方式采用基于版本信息的方法，即每一个文件标识符可能对应多个文件内容，具有同一个文件标识符的多个内容都被分配有一个不同的版本号。相应地，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ID、版本信息、数据分片内容和所有数据分片的指纹，分别用DID、Dversion、Dcontent和Dfingerprint表示。由于同一个DID可能具有多个不同版本的数据项，为方便对数据库中数据的检索，以DID和Dversion两个域组成数据的复合索引，其中DID作为第一索引，Dversion作为第二索引。初始化时，所有数据项的版本信息均为0；当针对某DID执行一次写操作时，新写入的数据项版本信息为该DID所对应的所有数据项中最高版本信息加1；针对某DID执行一次读操作时，存储服务器将该DID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。

具体的读写过程如下：

客户需要写文件F时，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、所在的组，以及存取模式和许可模式等，产生其目录路径等信息，同时产生并向客户返回一个文件标识符ID（该ID将作为各个存储服务器中存储文件F编码后数据分块时的第一索引）；客户在收到目录服务器返回的ID之后，向存储服务器集中的每个服务器发出以该ID为标志的写句柄请求；存储服务器集中的每个服务器在收到客户请求之后首先验证用户的身份及其权限，如权限允许，则查询以该ID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ID以及该ID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IDA编码方法对这m个文件分段进行编码，生成n个数据分块$D_1, D_2, \cdots, D_n$，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每个服务器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ID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。

类似地，客户需要读文件F时，首先向目录服务器发出请求，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、所在的组、存取模式和许可模式等信息判断是否向客户授权。如授权，则生成并向客户返回一个文件标识符ID；客户使用目录服务器返回的ID向服务器集发出读句柄请求；存储服务器集中的每个服务器在收到客户请求之后首先验证用户的权限，如权限允许，则查询以该分段ID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操作的版本信息，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集的每个服务器发出读请求；每个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ID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；客户对所接收到的数据项进行验证，如果存在来自r个存储服务器的数据项满足以下两条：（1）这r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IDA码译码方法，就可从这r个数据分块中恢复出原始文件。

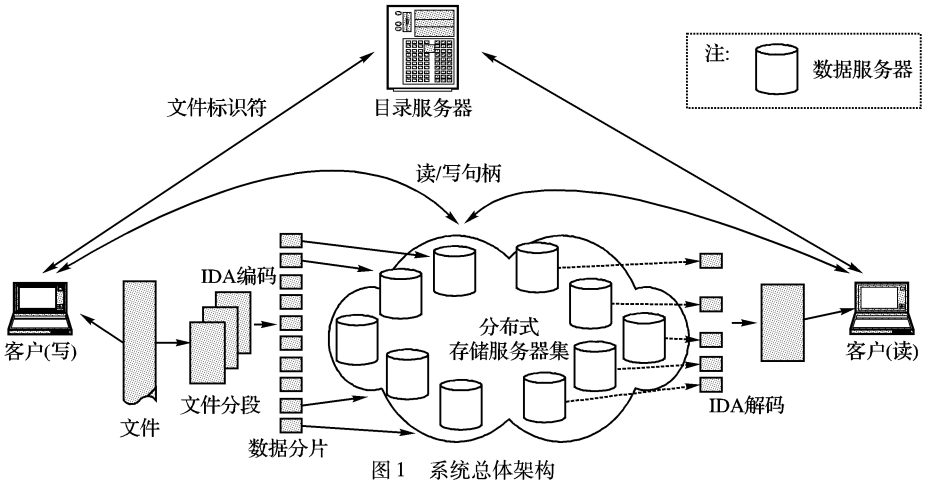

图1 系统总体架构

4　结语

本文提出的分布式系统中实用的容忍入侵数据存储方案,通过使用纠删码领域较新的一种编码方法——IDA 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。此外,通过基于数字指纹的安全 IDA 码存储方案设计,可满足恶意环境中分布式数据存储系统容忍 Byzantine 攻击者的要求。对系统方案的初步评估结果表明,该方案在系统的安全性与系统开销方面,有着较好的折中,具有较高的实用性。

参考文献:

[1] SCHNEIDER FB. Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial[J]. ACM Computing Surveys, 1990, 22(4): 299 - 319.

[2] CASTRO M. Practical Byzantine Fault Tolerance[D]. Technical Report MIT/LCS/TR-817. MIT Laboratory for Computer Science, Cambridge, MA, 2001.

[3] RODRIGUES R, LISKOV B. Rosebud: A Scalable Byzantine-Fault-Tolerant Storage Architecture[R]. MIT CSAIL Technical Report TR/932, 2003.

[4] AMIR Y, WOOL A. Optimal Availability Quorum Systems: Theory and Practice[J]. Information Processing Letters, 1998, 65(5): 223 - 228.

[5] GIFFORD DK. Weighted Voting for Replicated Data[A]. Proceedings of the seventh ACM symposium on Operating systems principles table of contents[C]. Pacific Grove, California, United States, 1979. 150 - 162.

[6] MALKHI D, REITER M, WOOL A. The load and availability of Byzantine quorum systems[A]. Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing[C]. Santa Barbara, California, United States, 1997. 249 - 257.

[7] MALKHI D, REITER MK. Persistent objects in the fleet system [A]. Proceedings of the 2nd DARPA Information Survivability Conference and Exposition (DISCEX II)[C], 2001.

[8] MARTIN JP, ALVISI L. A Framework for Dynamic Byzantine Storage[A]. 2004 International Conference on Dependable Systems and Networks (DSN 2004)[C]. Florence, Italy: IEEE Computer Society, 2004. 325 - 334.

[9] LAKSHMANAN S, AHAMAD M, VENKATESWARAN H. Responsive Security for Stored Data[J]. IEEE Transactions on parallel and distributed system, 2003, 14(9): 818 - 828.

[10] SUBBIAH A, BLOUGH DM. An Approach for Fault Tolerant and Secure Data Storage in Collaborative Work Environments[A]. Proceedings of the 2005 ACM workshop on Storage security and survivability[C], Fairfax, VA, USA, 2005.

[11] RABIN MO. Efficient dispersal of information for security, load balancing, and fault tolerance[J]. Journal of the ACM, 1989, 36(2): 335 - 348.

(上接第 1098 页)

工作机允许 op 为只读 r(01)时,只有迁移实例访问信息中 op 为 r(01)时,工作机提交产品目录,其余情况(例如 r/w(01/02),w(02)等)工作机全部拒绝提供目录服务。

迁移工作流管理系统涉及诸多不安全因素,原型测试表明,该安全模型能够较好的解决工作位置主机面临的安全问题。

图 5　安全加载界面

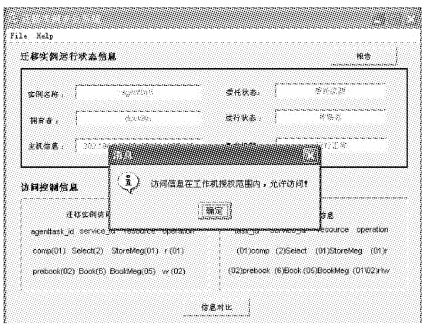

图 6　状态监测与访问控制界面

4　结语

本文根据迁移工作流管理系统自身的安全特点,提出了一个工作位置安全模型,并讨论了安全加载、状态监测和访问控制等关键技术和实现。由于迁移工作流管理系统安全的特殊性,还有很多问题等待进一步解决,例如,为了保证迁移域的可靠性,本文中迁移实例的迁移路径是管理机与工作位置事前商议好的,并且本文中没有涉及对于迁移实例本身如何进行保护的问题。如何组织一个良好的工作位置信任关系,建立迁移实例信任域,实现安全的迁移实例动态迁移;如何实现在工作位置上的迁移实例自保护问题,这些是下一步研究的重点问题。

参考文献:

[1] CICHOCKI A, RUSINKIEWICZ M. Migrating workflows [A]. DOGAC A, ed. Workflow Management System and Interoperability [C]. Berlin: Heidelberg(Spring Verlag), 1998. 339 - 355.

[2] 李洪霞, 王晓琳, 曾广周. 迁移工作流中的自适应信任模型[J]. 计算机应用, 2003, 23(11): 97 - 99.

[3] 曾广周, 党研. 基于移动计算范型的迁移工作流研究[J]. 计算机学报, 2003, 26(10): 1343 - 1349.

[4] WU SL, SHELTH A, MILLER J, *et al.* Authorization and access control of application data in workflow systems[J]. Journal of Intelligent Information Systems, 2002, 18(1): 71 - 94.

[5] KANDALA S, SECURE SR. Role _ based workflow models [A]. Proceedings of the 15th IFIP WG11. 3 Working Conference on Database Security[C]. Kluwer, 2002.

[6] 邓集波, 洪帆. 基于任务的访问控制模型[J]. 软件学报, 2003, 14(01): 76 - 82.

[7] TSCHUDIN CF. Mobile Agent Security[A]. KLUSCH M, ed. Intelligent information agents: Agent based information discovery and management in the Internet[C]. Berlin: Springer-Verlag, 1999. 431 - 446.

[8] http://www.fipa.org/specs/fipa00020/OC00020A.pdf[EB/OL].

[9] 杨鲲, 刘大有, 郭欣. 一个具有高安全性的移动 Agent 系统模板结构[J]. 软件学报, 2002, 13(1): 130 - 135.