HDFS 的多安全级数据销毁机制设计

秦 军[1],邓 谦[2],张建平[2]
(1. 南京邮电大学 教育科学与技术学院,江苏 南京 210003;
2. 南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:在云计算应用中,数据安全是用户首要关心的问题,因此云中数据安全的研究也成为当前云计算研究的重点。针对开源云计算存储系统 HDFS 中的数据不能彻底销毁,从而可能导致数据泄露的问题,设计了 HDFS 的多安全级数据销毁机制。一方面,该机制在删除数据前使用数据覆写算法覆写原数据,可以有效预防云中数据的恶意恢复,防止数据泄露,从而达到彻底销毁数据的目的;另一方面,该机制采用多安全级可定义的方法,采取多种覆写算法销毁数据,平衡了安全需求和性能需求。仿真实验表明,该机制可以在 HDFS 环境下有效地覆写 Block 文件达到彻底销毁原始数据的目的,同时不同的覆写算法时间开销也不同,保证了效率和安全的平衡。

关键词:云计算;数据安全;HDFS;数据覆写;多安全级
中图分类号:TP311.133.1 **文献标识码**:A **文章编号**:1673-629X(2013)03-0129-05
doi:10.3969/j.issn.1673-629X.2013.03.033

Design of Multi-grade Safety Data Destruction Mechanism of HDFS

QIN Jun[1], DENG Qian[2], ZHANG Jian-ping[2]
(1. College of Education Science and Technology, Nanjing University of Posts & Telecommunications, Nanjing 210003, China;
2. College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract: In the cloud computing applications, data security is the first concern to users, so the data security, in the cloud, becomes the current research focus. The open source cloud computing storage system HDFS can't completely destroy data, which may lead to data leak. To solve this flaw, design a multi-grade safety data destruction mechanism of HDFS. This mechanism can effectively prevent malicious data recovery and prevent data leak, destroy data completely by overwriting original data with overwrite algorithm before deleting data. Moreover, the mechanism can balance the safety requirement and performance requirement by using different overwrite algorithm according to different safety requirement. The simulation experiment shows that the mechanism can effectively overwrite the file of Block to destroy original data completely in HDFS. And the spending time of different overwrite algorithm is different, so it can keep efficiency and safety balance.

Key words: cloud computing; data security; HDFS; data overwrite; multi-grade safety

0 引 言

随着云计算的兴起,很多公司开始对外提供云计算服务[1~3],如 Google 的 GAE[4] 平台,Amazon 的弹性云计算平台 AWS[5] 等。除了这些封闭的云计算系统,还有许多公司选择使用开源的 Hadoop 系统搭建自己的云平台。Hadoop[6] 是由 Apache 基金会支持的一个开源的云计算系统,其核心项目由 Common、MapReduce 和 HDFS(Hadoop Distributed File System)组成。其中 HDFS[7] 是为整个 Hadoop 以及相关项目提供数据分布式存储的文件系统。

根据 IDC 在 2008 年发布的一项调查报告显示[8],安全性是人们对云计算最为关心的问题,其中数据的安全性是用户关心的主要问题。目前的云系统中,数据的安全性、可靠性还存在着某些的问题[9]。也正是出于安全性的考虑,许多小公司不愿将数据迁移到云中。同时,由于很多公司使用 Hadoop 系统开发云计算平台,如 Yahoo 的数据处理与存储集群,阿里巴巴、百度等开发的云计算平台,HDFS 存储系统中的数据安全和可靠性都是人们关心的一个主要问题。特别是现在的 HDFS 对于数据的安全删除却没有一个有效的机

收稿日期:2012-06-19;**修回日期**:2012-09-23
基金项目:江苏省自然基金项目(BK2009425);江苏省教育科学"十二五"规划课题(D/2011/01/074)
作者简介:秦 军(1955-),女,教授,硕士生导师,主要研究方向为计算机网络技术、多媒体技术、数据库技术;邓 谦(1987-),男,河北石家庄人,硕士研究生,主要研究方向为分布式计算、云计算。

制,恶意用户以及超级管理员可以使用恢复软件恢复删除后的数据,这将会是对数据保密性的一个巨大威胁。针对用户数据在云中所面临的数据泄露的问题,尤其是删除的数据会被恶意恢复的问题,文中提出了一种多安全级可定义的数据销毁机制,通过使用覆写算法,将要删除的数据完全覆盖,可以有效提高数据在 HDFS 中的安全性,避免任何不怀好意者恢复和窃取用户已经删除了的数据。

1 HDFS 的数据安全问题

Gartner 于 2008 年发布的《评估云计算安全风险》中列举了云计算的 7 大风险[8],其中涉及数据安全的就有三大风险:数据隔离风险、数据恢复风险、数据位置风险。数据安全问题是云计算所面对的主要问题。同时,由于云系统的分布式特性、多用户特性、用户数据的所有权与管理权的分离,云系统中的数据会面临许多新的其他方面风险。

针对数据的完整性、数据的可用性问题,HDFS 通过用户身份认证[10]、文件校验和、数据的容灾备份等机制提供了较好的保护。但是针对数据的保密性,尤其是针对删除后的数据没有提供有效保护,没有提供一个安全有效的数据删除机制。

数据的不安全销毁在单机环境下或者私有集群中,对于一般用户不是很严重的问题,因为用户对于存储设备拥有所有权,可以实现数据的物理隔绝,杜绝任何非法用户恢复删除的数据。但是当用户将数据和计算迁移到云中后,尤其是公有云中,用户就失去了对存储设备的所有权,就不能阻止不怀好意者试图恢复用户删除的数据,从而窃取用户数据。由于云计算的多用户属性以及超级用户权限的存在,如果用户不能安全有效地删除数据,那么这些数据就有可能被那些不怀好意的用户或者是系统的管理者通过软件或者物理手段恢复出来。这对于用户的一些敏感的数据是巨大的威胁。

2 HDFS 结构分析

2.1 HDFS 数据存储机制

HDFS 整个存储体系是由 NameNode、DataNode 和客户端三部分组成[11]。HDFS 的数据存储流程如下:

1)客户端与 NameNode 通信,申请创建文件。

2)当 NameNode 准许用户创建文件后,会生成这个文件的元数据(MetaData)并将这些元数据保存在本地的文件系统中,同时将文件分成数个 Block,每个数据块默认值是 64M。这些 Block 将分别存储在不同的 DataNode 上,NameNode 将维护这个文件到各个 Block 的映射。

3)当元数据建立好后,NameNode 将通知客户端与相应 DataNode 通信,直接将数据写到 DataNode 上。

4)DataNode 将接收到的 Block 以单个文件的形式保存到本地的文件系统上。DataNode 并不知道 HDFS 的文件组织,其只维护在本地文件系统中的数据。

2.2 HDFS 数据的删除机制

HDFS 中数据的删除是异步的。在 Namenode 将元数据删除后,Datanode 通过 heartbeat 获取此改变(invalid block 列表),并将其对应的文件删除。

图 1 和图 2 分别显示了 Namenode 和 Datanode 与文件删除相关的类图。

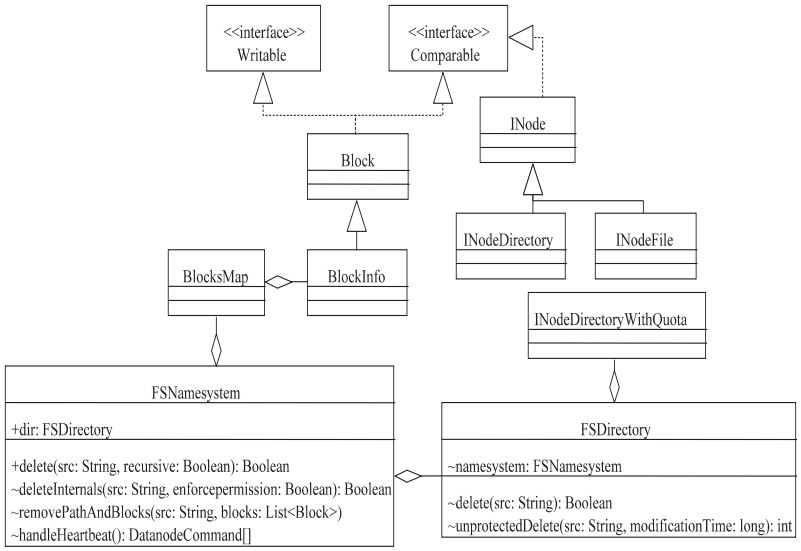

图 1 Namenode 数据删除类图

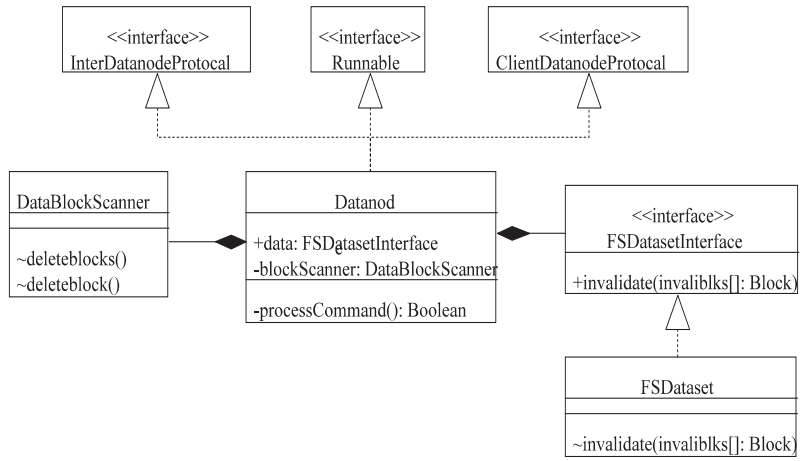

图 2 Datanode 数据删除类图

客户端流程:ClientProtocol 的 delete( ) 方法 RPC 远程调用 Namenode 同名方法。

●Namenode 端流程:

1)在 Namenode 中调用 FSNameSystem 的 delete( ) 方法,该方法首先会检测删除是否为递归删除,如果是递归删除,则采用递归删除,删除目录及目录中的内容。如果不是递归删除,则检查删除的目录是否为空,如果目录不为空则抛出异常;

2)FSNameSystem 的 delete( ) 方法随后调用 delete Internal( ) 方法,该方法主要是进行安全模式检测和权限检查,并调用 FSDirectory 的对象 dir 的 delete( ) 方法;

3)FSDirectory 的 delete( ) 方法主要是通过调用 unprotectedDelete( ) 方法来删除 namespace 中的元数据,并向日志写入删除记录;

4)unprotectedDelete( ) 方法将 inode 从 namespace 中删除,然后调用 FSNameSystem 的对象 namesystem 的 removePathAndBloc( ) 方法;

5) removePathAndBlocks( ) 方法首先移除租约(lease),然后将要删除文件的数据块对应的 inode 从 block 中移除并将 block 从 CorruptBlocksMap 中移除,最后将要删除的 block 加入到失效块列表(list of blocks which will be invalidated)中;

6)当 Datanode 向 Namenode 发送 heartbeat 时,FSNameSystem 中 handleHeartBeat( ) 会将 cmds 数组中的一个指令设置为 blockInvalidateLimit 这个常量用以指示 Datanode 删除相应的 blocks,并将需要删除的块列表,包含到 BlockCommand 中,返回给 Datanode。

●Datanode 端流程:

1)当 Datanode 收到删除命令的时候,首先是调用 DataBlockScanner 类中的 deleteBlock 方法删除 Datanode 中相应的元数据;

2)调用 FSdataset 中的 invalidata( ) 方法,在 invalidate( ) 方法中会对要删除的 Blocks 进行一些异常检测,并将 block 从 volumemap 中删除,最后调用 java 的 File 类对象 f. delete( ) 将数据删除。

通过分析 HDFS 数据的存储和删除机制,可以得知,HDFS 的 Block 是存储到 Datanode 的本地文件系统上的,并通过 java 的文件删除机制删除。由此可见其删除机制与单机状态下普通的文件删除是相同的。

Linux 系统的 ext2 文件系统的文件删除机制同样只是将块位图和索引节点的状态改变了[12],导致系统不能通过索引节点去定位文件,但是文件的实际内容甚至是索引节点中的元数据都没有真正删除,只有等下一次数据写入的时候才能覆盖这些数据。所以,HDFS 中的数据并没有被实际删除,这在所有权和管理权分离的分布式环境中是一个很严重的安全隐患。别有用心者可以通过数据恢复技术,恢复出用户删除的数据,而用户对这一过程一无所知,也完全无法防范。

2.3　HDFS 的异构存储环境

HDFS 系统是一个异构的分布式文件系统,其可以支持异构的主机构成一个统一的文件系统。其中,存储的异构特性主要体现在存储介质的不同,现在主流存储介质有传统硬盘的磁介质和固态硬盘(SSD)的闪存半导体介质。这两种存储介质由于数据存储原理不同,针对其采用的数据销毁技术也不同。

磁介质的存储设备主要是通过电磁原理来存取数据。在一个扇区中,磁头通过磁化每个磁粒子来存储数据,每个磁极表示一个 0 或者 1 状态。每个扇区可以存储 4096 个这种状态,也就是 512B。由于磁化后的边缘残留和不完全磁化的影响,通常一遍覆盖并不能完全覆盖数据,通过专业的设配通过分析硬盘还是能将覆盖前的数据恢复。所以选择有效的覆写方法、覆写次数,是保证磁盘数据不被恢复的关键。

固态硬盘通过半导体存储介质来存储数据。现在的主流固态硬盘是采用 Flash 介质,通过叫做“浮动门场效应晶体管”的晶体管来保存数据。这样的每一个晶体管叫做一个 cell,每个 cell 是通过电荷的充放电来表示数据的 0 和 1[13]。由于固态硬盘的存储原理不同于磁盘,并不会存在剩磁效应,所以并不需要使用特定的序列进行多次覆写。

3　HDFS 多安全级数据销毁机制

针对 HDFS 不能彻底删除用户存储的数据的缺陷,文中设计了 HDFS 的多安全级数据销毁机制,以达到数据的安全销毁。

3.1　数据销毁技术

目前的数据销毁技术大致可以分两类:硬销毁技术和软销毁技术[14]。硬销毁技术主要包括消磁技术、物理销毁技术等。硬销毁技术主要是通过对存储介质的永久性破坏来达到安全要求。虽然硬销毁技术能绝对保证删除的数据安全,但是因为完全破坏了存储介质,导致设备不能重复使用,这对于 HDFS 是不能接受的,也是基本不能实现的。

软销毁技术,主要是通过软件的方法删除数据。软销毁技术中的数据覆盖技术可以通过采用特定的覆写规则和覆写序列覆盖存储介质上的原有数据。虽然数据覆盖技术在进行一次覆盖后,由于磁存储介质的剩磁效应,并不能将数据彻底删除。但是通过增加覆写次数和调整覆写规则,可以在很大程度上销毁数据,防止数据被非法恢复。对于 SSD 等闪存类存储器,因

为没有剩磁效应,通过一次覆写就可以达到数据销毁效果。所以文中采用了软销毁技术中的数据覆盖技术。

目前,主流的覆写标准有 DOD5220.22-M[15] 简单覆写标准、DOD5220.22-M 7 次擦除标准、全零覆写标准、RCMP TSSIT OPS-II[16] 标准和 Gutmann[17] 等标准。不同的标准安全性不同,同时消耗的资源和时间也不相同。

3.2 数据销毁流程

图 3 给出了 HDFS 多级安全级数据销毁流程图。

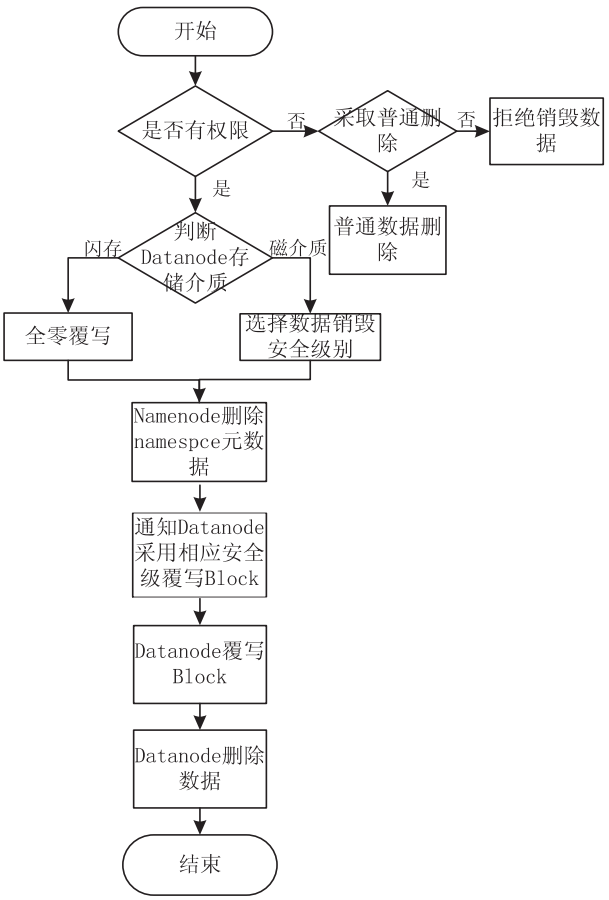

图 3 HDFS 多安全级数据销毁流程图

整个流程和原有的 HDFS 数据删除流程的主要区别是:

1)加强的权限认证机制。在进行数据删除之前要首先根据用户的权限判别用户是否有权进行数据销毁,如果用户没有数据销毁权限,将询问用户是否使用普通的数据删除。这个过程主要是预防恶意用户通过使用高安全级别的数据删除策略,恶意覆写大规模数据,导致系统资源被过度使用;

2)根据 Datanode 的存储介质判断使用的数据删除策略,而不是使用统一的数据删除方法。由于固态硬盘没有剩磁效应,所以固态硬盘使用全零覆写标准覆写一次就可以达到数据销毁的目的。磁介质则采取多次数据覆写标准,如 DOD5220.22-M 等。这样可以使整个 HDFS 数据删除的效率更高,不会造成系统资源的浪费;

3)根据用户的安全需求,采取不同的数据覆写策略。由于不同用户对数据销毁的安全等级需求不同,或者是同一个用户对不同数据的销毁等级需求不同,所以多安全级的数据销毁机制可以满足不同用户的不同需求,如当用户删除低敏感度数据的时候可以使用 DOD5220.22-M,在删除高敏感度数据的时候采用 DOD5220.22-M 7 次擦除标准。采用不同的安全级别的数据删除策略,可以在有效销毁数据的同时避免由于统一采用高安全级别造成的系统资源大量占用的情况出现。

3.3 覆写算法

文中采用的覆写算法有:1)全零覆写,全零覆写只是用在固态硬盘的数据删除上;2)DOD5220.22-M 覆写标准;3)DOD5220.22-M 7 覆写标准;4)RCMP TSSIT OPS-II 标准。在设计中并没有采用 Gutman 覆写标准。虽然该算法现在是最安全的覆写算法,但是由于要覆写 35 次之多,覆写速度慢,时间长。同时由于 HDFS 存储的数据都是大数据量数据,很大部分是 GB 级甚至 TB 级以上的数据,若采用 Gutman 覆写算法其效率太低,将会对系统的性能造成很大的影响。

覆写算法如下[14]:

●算法 1 全零覆写算法

往文件中全部覆写一次 0。

●算法 2 DOD5220.22-M 覆写算法

1)产生一个随机数,用该随机数覆写文件。

2)取该随机数的反码,用该反码覆写文件。

3)生成另一个随机数,用该随机数覆写文件。

●算法 3 DOD5220.22-M7 覆写算法

1)产生一个随机数,用该随机数覆写文件。

2)取该随机数的反码,用该反码覆写文件。

3)产生另一个随机数,用该随机数覆写文件。

4)产生另一个随机数,用该随机数覆写文件。

5)取该随机数的反码,用该反码覆写文件。

6)产生另一个随机数,用该随机数覆写文件。

7)产生另一个随机数,用该随机数覆写文件。

●算法 4 RCMP TSSIT OPS-II 覆写算法

该覆写算法一共覆写 8 次,奇数次产生随机数并用该随机数覆写文件,偶数次用上次随机数的反码覆写文件。

4 实验结果与性能分析

不同的覆写算法的安全性和性能在许多文章中都已经得到了验证,文中的实验主要是验证在 HDFS 环境中覆写算法对 Block 覆写的有效性和性能。

4.1　实验环境和工具

实验环境如表 1 所示,因为 Block 存储在 Datanode 的本地文件系统中,并不涉及分布式环境,所以将 Hadoop 设置为单机伪分布模式。

表 1　实验环境

物理机	虚拟机	Hadoop 配置
CPU:Intel Core DuoT5250 1.5GHz 内存:2GB 硬盘:WD 120G 8MB 缓存 5400 转/分	CPU:单核 内存:1GB 操作系统:CentOS 5.5	版本:Hadoop-0.21.0 配置:单机伪分布模式

仿真程序采用自主编写的 Java 软件,利用 FileOutputStream 类和 BufferedOutputstream 类实现文件的覆写,经过多次验证,软件可以有效地覆写目标文件。

4.2　实验结果

实验将 60M 的文本文件存储在 HDFS 中,产生一个 64M 的 Block,并对这个 Block 的存储文件采取不同安全级的覆写算法进行覆写。如图 4 所示,安全级越高的覆写算法,时间开销越大。覆写过后的文件,在 HDFS 中已经无法正确读取,结果如图 5 所示。

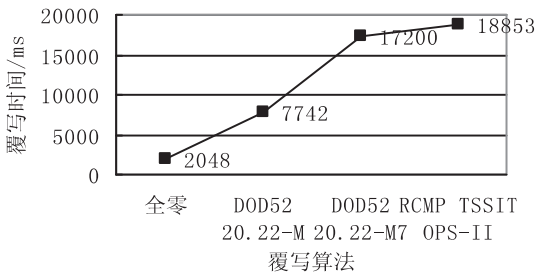

图 4　验证不同算法覆写性能

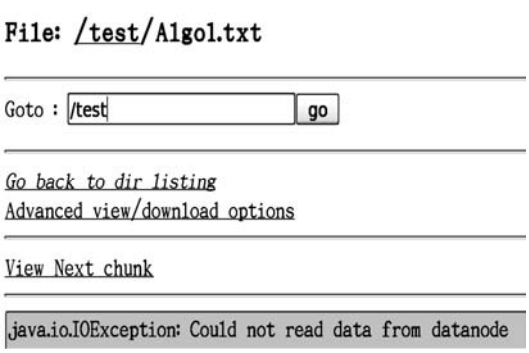

图 5　覆写后的 HDFS 文件

5　结束语

针对 HDFS 文件系统中的数据销毁机制存在的缺陷,文中提出了 HDFS 的多安全级数据销毁机制。云存储是将来的主要存储方式,越来越多的数据将会迁移到云端。虽然针对云中数据安全提出了许多安全措施,但是很少有措施是针对云中数据销毁问题的。文中将数据覆写技术应用到了 HDFS 的数据删除机制中,彻底地销毁了数据,有效地防范了恶意恢复删除的数据。该机制的权限检测预防了恶意用户使用覆写机制造成的系统资源占用,同时安全多级可定义特性有效地平衡了安全需求和性能需求。

接下来的工作主要是完善身份认证机制,防止恶意删除,以及开发更适用于分布式体系的数据覆写机制,以提高数据删除的效率。

参考文献:

[1] 陈　康,郑维民. 云计算:系统实例与研究现状[J]. 软件学报,2009,20(5):1333-1348.

[2] 王德政,申山宏,周宁宁. 云计算环境下的数据存储[J]. 计算机技术与发展,2011,21(4):81-84.

[3] 刘　鹏. 云计算[M]. 北京:电子工业出版社,2010:69-70.

[4] Google,Inc. What is google app engine? [EB/OL]. [2012-03-30]. http://zh.wikipedia.org/wiki/Google_App_Engine.

[5] 张建成,宋丽华,鹿全礼,等. 云计算方案分析研究[J]. 计算机技术与发展,2012,22(1):165-167.

[6] The Apache Software Foundation. Welcome to ApacheTM HadoopTM! [EB/OL]. [2012-03-30]. http://hadoop.apache.org/skin/images/pdfdoc.gif.

[7] The Apache Software Foundation. HDFS users guide[EB/OL]. [2012-03-30]. http://hadoop.apache.org/common/docs/current/hdfs_user_guide.pdf.

[8] 中国电信网络安全实验室. 云计算安全:技术与应用[M]. 北京:电子工业出版社,2012:12-20.

[9] 张　慧,邢培振. 云计算环境下信息安全分析[J]. 计算机技术与发展,2011,21(12):164-166.

[10] 柴黄琪,苏　成. 基于 HDFS 的安全机制设计[J]. 计算机安全,2010(12):22-25.

[11] DHRUBA BORTHAKUR. HDFS architecture guide[EB/OL]. [2012-04-03]. http://hadoop.apache.org/common/docs/current/hdfs_design.pdf.

[12] 冯　锐,王　磊. 如何恢复 Linux 上删除的文件,第 3 部分[EB/OL]. [2012-04-04]. http://www.ibm.com/developerworks/cn/linux/l-cn-filesrc3.

[13] 张　冬. 大话存储 II[M]. 北京:清华大学出版社,2011:50-51.

[14] 程　玉. 磁介质数据销毁技术的研究[D]. 成都:电子科技大学,2010.

[15] Wikipedia. National Industrial Security Program[EB/OL]. [2012-04-05]. http://en.wikipedia.org/wiki/DOD_5220.22-M.

[16] Tim Fisher. RCMP TSSIT OPS-II [EB/OL]. [2012-04-05]. http://pcsupport.about.com/od/termsr/g/rcmp-tssit-ops-ii.htm.

[17] Wikipedia. Gutmann method[EB/OL]. [2012-04-05]. http://en.wikipedia.org/wiki/Gutmann_method.

HDFS的多安全级数据销毁机制设计

作者： 秦军， 邓谦， 张建平
作者单位： 秦军(南京邮电大学 教育科学与技术学院,江苏 南京210003)， 邓谦,张建平(南京邮电大学 计算机学院,江苏 南京210003)

刊名： 计算机技术与发展
英文刊名： Computer Technology and Development
年，卷(期)： 2013(3)