

分布式个人文件系统的设计与实现

何兴高, 张凤荔, 黄远军, 秦志光, 周明天

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】提出了一种基于E-mail系统的分布式文件系统——EMDFS, 给出了扩展的SMTP协议(ESTMP)的状态转换方式和定义, 在此基础上研究了利用ESMTP来构建分布式个人文件系统的方法和模型, 设计了EMDFS的模型、内外存的结构、I/O操作、用户接口以及EMDFS的各种功能。

关键词 简单邮件传输协议; 互联网消息存取协议4; 个人网络存储; 分布式文件系统

中图分类号 TP393 文献标识码 A

Design and Implementation of Distributed Personal File System

HE Xin-gao, ZHANG Feng-li, HUANG Yuan-jun, QIN Zhi-guang, ZHOU Ming-tian

(School of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract This paper presents a distributed file system based on E-mail named E-mail distributed file system. This paper gives the state and definition of extension SMTP, based on this we design the model and method of the EMDFS, and propose the store space, the memory and disk structure of EMDFS, the I/O operators, user interface, and other functions.

Key words simple mail transfer protocol; internet message access protocol-version 4; person net ware storage; distributed file system

本文提出了一种基于分布式环境的个人数据的网络存储方式, 对现有的网络协议进行扩充, 利用E-mail, 解决个人数据文件在分布式网络环境下的实时存储、共享。

1 E-mail协议及其扩展

E-mail协议包括简单邮件传输协议(Simple Mail Transfer Protocol, SMTP)^[1], 简单邮件传输协议服务扩展(Extended SMTP: ESMTP)^[2], 邮局协议3(Post Office Protocol-Version 3, POP3), 互联网消息存取协议4(Internet Message Access Protocol-Version 4, IMAP4)^[3], 多用途网际邮件扩展(Multipurpose Internet Mail Extensions, MIME)^[4]。SMTP本身没有存储空间的概念, 对SMTP进行存储扩展, 就要引入个人存储空间扩展的概念(Storage extended SMTP, SSMTP)。默认的个人存储空间是SMAILBOX; 引入SMAILBOX, 可避免普通邮件同个人网络存储的数据相混淆。SSMTP连接后, 进入普通的SMTP状态(Non-SSMTP状态), 进行邮件操作。用户可以使用特殊命令SHLO, 切换到SSMTP个人存储空间。为了保护用户个人空间, 必须对用户进行身份验证, 验证成功后, 选择个人空间进入; 消息发送和个人数据的就以消息格式存储在一条消息中, 包含个人数据的所有的消息, 都存储在该个人存储空间中。SSMTP协议包括Non-SSMTP状态、

收稿日期: 2004-06-08

基金项目: 四川省科技攻关项目(104JY029-001-3)

作者简介: 何兴高(1964-), 男, 硕士, 工程师, 主要从事计算机控制、智能交通系统方面的研究。

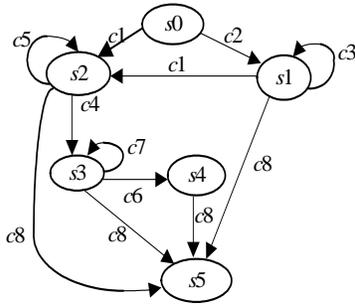


图1 SSMTP状态转换图

Non-Authenticated状态、Authenticated状态、Selected状态和Logout状态5个状态，如图1所示。有限状态机(Finite State Machine, FSM)模型是这些状态之间的状态迁移图。

引入一个初始状态 $s_0 = \text{SSMTP}$ 连接初始建立状态，即有限状态机可以表示为：

$$M = (S, \Sigma, f, s_0, Z)$$

式中 S 是SSMTP系统所有状态的集合， $S = \{s_0, s_1, s_2, s_3, s_4, s_5\}$ ， $s_1 = \text{Non-SSMTP}$ 状态(SMTP系统的初始状态，连接初始建立状态)， $s_2 = \text{Non-Authenticated}$ 状态， $s_3 = \text{Authenticated}$ 状态， $s_4 = \text{Selected}$ 状态， $s_5 = \text{Logout}$ 状态(SSMTP系统的终态集)； Σ 是SSMTP所有可以引起系统状态改变的命令或者系统事件， $\Sigma = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ ， $c_1 = \text{SHLO}$ 成功， $c_2 = \text{HELO+SHLO}$ 失败， $c_3 = \text{SHLO}$ 失败， $c_4 = \text{AUTHENTICATE}$ 成功， $c_5 = \text{AUTHENTICATE}$ 失败， $c_6 = \text{SELECT}$ 成功， $c_7 = \text{SELECT}$ 失败， $c_8 = \text{QUIT+服务器断开+连接关闭}$ ； f 为转换规则， $f(s_i, c_k) = s_j \quad i, j, k = 0, 1, 2, \dots$ ； Z 是结束状态， $Z = s_5$ 。

2 基于E-mail的分布式文件系统模型

文件系统分为三个层次：1) 最低层是对象及其属性说明；2) 中间层是对对象进行操作和管理的软件集合；3) 最上层是文件系统。

分布式文件系统是文件系统向网络扩展的结果。在分布式文件系统中，文件、目录不再是存储在单独的计算机外存上，而是分布在网络连接起来的若干台计算机外存上，磁盘的读写操作变成了对网络接口的读写。

基于E-mail的分布式文件系统(E-Mail Distributed File System, EMDFS)模型如图2所示。EMDFS的对象包括文件、目录和E-mail邮箱空间。在分布式文件系统中，存储空间构架在若干E-mail邮箱空间，并无地理空间的限制，还可以运行在不同的硬件平台和不同的操作系统平台。逻辑文件系统是文件系统在内存中的表现形式和与E-mail有关的操作，包括文件在多个E-mail邮箱中的组织、E-mail的发送和接收操作等。

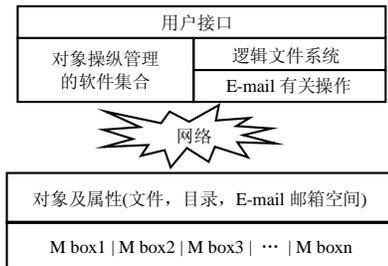


图2 EMDFS模型

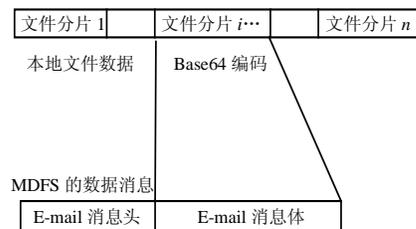


图3 本地文件转化成EMDFS

3 EMDFS的对象及其属性说明

EMDFS文件属性包括文件名、文件属主、文件大小、创建时间、所在的E-mail邮箱等。EMDFS采用单级目录结构，EMDFS存储空间由若干个E-mail邮箱空间构成，每一个E-mail邮箱空间可被看成是EMDFS的一块磁盘分区。在E-mail邮箱中，E-mail消息是存储的元素，E-mail消息的结构由RFC822、RFC2045、RFC2046规定。EMDFS存储的单位是E-mail消息，EMDFS中的E-mail消息对应于文件系统中磁盘扇区的概念。EMDFS的存储消息分为存储E-mail邮箱信息的超级消息和存储文件数据的数据消息两类。本地文件数据被切分成若干文件分片，每一个文件分片被封装成一条E-mail消息。文件分片被存储在一个或多个E-mail邮箱，通过数据消息的属性之间的联系，实现分布式存储。

超级消息是EMDFS存储空间的属性信息，保存了构成EMDFS的存储空间的E-mail邮箱信息，用来访问

E-mail邮箱。超级消息在EMDFS的每一个E-mail邮箱保存一个副本, 当有新的E-mail邮箱加入到EMDFS的存储空间中, 或者某个E-mail邮箱被从EMDFS存储空间删除时, 需要更新EMDFS的超级消息。超级消息由消息体和消息头构成。消息体保存了构成EMDFS的若干E-mail邮箱消息体保存了构成EMDFS的若干E-mail邮箱信息。超级消息的消息体=Base64(邮箱1信息 + 邮箱2信息+ ... + 邮箱n信息)。邮箱信息 = 邮箱地址*空间容量*SMTP地址*POP3或IMAP地址*用户名*密码。一条邮箱信息对应于一个E-mail邮箱, EMDFS可以通过超级信息, 访问每一个EMDFS的E-mail邮箱。文件在EMDFS上都是以消息的形式存储的。图3是本地文件转化成EMDFS的数据消息的图示, 消息体是文件分片数据, 采用Base64编码。消息头的作用是用来标志消息是文件分片数据、作为文件分片重新还原成文件的索引信息, 表1是个人分布式网络存储系统中消息头结构的域及其含义。

表1 基于E-mail系统的个人分布式网络存储的消息头结构定义

域名	特定格式	定义
Date	Date: 星期, 日 月 年 时:分:秒	文件分片消息创建的时间
From	From: 电子邮箱地址	邮件发送者地址, 服务器上的邮箱地址
To	To: 电子邮箱地址	邮件接收者地址, 服务器上的邮箱地址
X-MDFS	X-MDFS: MDFS	基于E-mail的分布式文件系统标志:MDFS
X-Filefragment	X-Filefragment: 使用Base64编码的分片属性信息	文件分片属性信息:文件名*文件大小*文件分片数*分片字节数*分片编号*分片在文件中的开始地址
X-Smailboxs	X-Smailboxs:邮箱1*邮箱2*...*邮箱p	文件存储的E-mail空间列表;邮箱之间使用“*”连接
MIME-Version	MIME-Version: 1.0	MIME的版本信息, 采用1.0版本
Content-Transfer-Encoding	Content-Transfer-Encoding: Base64	消息体采用的编码算法:Base64

4 EMDFS的逻辑文件系统

EMDFS的逻辑文件系统包括EMDFS的内存结构和用户接口。内存维护EMDFS的分区列表(邮箱列表)、文件分片列表、文件列表。邮箱列表的每一个节点被称为M-node, M-node结构定义为struct M_node{ struct MBOX Mail_box; struct M_node *next_mailbox }; 文件分片列表的每一个节点被称为FF-node; 文件列表的每一个节点被称为F-node。EMDFS的内存结构采用链表形式, 如图4所示。有了EMDFS的逻辑结构, EMDFS的命令接口有更新超级消息, 创建文件, 读取文件, 删除文件等; 程序接口是提供给程序的库函数, 包括更新超级消息、创建文件、读取文件、删除文件等。

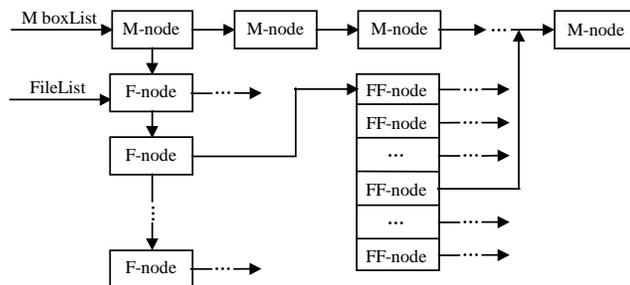


图4 EMDFS在内存中的结构

FF-node结构定义:

```
struct FF_node{
int FileFragmentID; //分片编号
int FileFragmentSize; //分片原始数据字节数
int FileFragmentAddress; //分片原始数据在文件中的开始地址
char *FileFragmentTitle; //Base64编码分片在E-mail邮箱中检索
struct M_node *m_node;; //邮箱的M_node节点
}
```

F-node结构定义:

```
struct F_node{
char * FileName; //文件名
char * FileDate; //文件创建日期
int FileSize; //文件字节数
int FileFragmentAmount; //文件分片数
char * mboxs; //所在邮箱的串, 参考用
struct FF_node * ff_nodes; //指向文件所有FF_node的首地址指针
struct F_node * next; //指向下一个F_node的指针
}
```

5 EMDFS的逻辑文件系统的实现

对EMDFS的实现, 选用Windows平台, 并采用10个人E-mail邮箱作为EMDFS的存储空间。整个系统包括EMDFS的底层操作、I/O操作、命令和程序接口。底层操作包括: 1)发送一条E-mail消息, 用SMTP协议(及其扩展)发送消息(Send Message, SM); 2)读E-mail邮箱可用空间大小, 用POP3或IMAP, 记为接收保持存储大小(Receive Remain Storage Size, RRSS); 3)读取E-mail邮箱中的消息条数, 用POP3或IMAP, 记为接收消息数量(Receive Message Amount, RMA); 4)读取一条E-mail的消息头, 使用POP3或IMAP, 接收消息头(Receive Message Head, RMH); 5)读取一条E-mail消息的消息体, 使用POP3或IMAP, 记为接收消息体(receive message body, RMB); 6)删除一条E-mail消息, 使用POP3或IMAP, 记为删除消息>Delete Message, DM)。EMDFS的I/O操作如表2所示。

表2 EMDFS的I/O操作

I/O操作名称	函数名	功能	返回值
超级消息I/O读 操作	int MDFS_SUPER_IO_read(struct MBOX Mbox; char *Message)	从Mbox.E_mailAddress邮箱中读取超级消息。将消息存放在Message中。	0成功, -1失败。
超级消息I/O写 操作	int MDFS_SUPER_IO_write(struct MBOX Mbox;char *Message)	将超级消息Message发送到Mbox.E_mailAddress邮箱中。	0成功, -1失败。
超级消息I/O删 除操作	int DFS_SUPER_IO_delete(struct MBOX Mbox;)	删除Mbox.E_mailAddress邮箱中的超级消息	0成功, -1失败。
数据消息I/O读 操作	int MDFS_DATA_IO_read(struct MBOX Mbox;struct FILEFRAGMENT FileFragment;char *Message)	从Mbox.E_mailAddress邮箱读取文件名为FileFragment.FileName编号FileFragment.FileFragmentID的数据消息。将分片消息保存在Message中。	0成功, -1失败。
数据消息I/O写 操作	int MDFS_DATA_IO_write(struct MBOX Mbox;char *Message)	将数据消息Message发送到MessageMbox.E_mailAddress邮箱。	0成功, -1失败。
数据消息I/O删 除操作	Int MDFS_DATA_IO_delete(struct MBOX Mbox; struct FILEFRAGMENT FileFragment)	删除Mbox.E_mailAddress邮箱中文件名为FileFragment.FileName的第FileFragment.FileFragmentID号分片。	0成功, -1失败。

与通常的文件系统相比, EMDFS具有主要应用于个人网络存储、实现在应用层、外存为E-mail邮箱和构建于若干个E-mail邮箱空间等特点。

6 结束语

以上实验表明, EMDFS可以方便、快捷地完成个人网络存储, 系统的响应时间在测试环境中是可以接受的。但对基于E-mail的分布式文件系统EMDFS的数据安全性、数据在E-mail空间存储的优化, 以及数据的一致性、高速缓存、文件锁、文件共享和多级目录等问题, 还需继续进行探讨。

参 考 文 献

- [1] Postel J B. Simple mail transfer protocol[DB/OL]. <http://www.isi.edu/in-notes/rfc821.txt>, 2004 - 03 - 05
- [2] Klensin J. SMTP service extension[DB/OL]. <http://www.isi.edu/in-notes/rfc1425.txt>, 2004 - 03 - 20
- [3] Crispin M. Internet message access protocol-version 4rev1 [DB/OL]. <http://www.isi.edu/in-notes/rfc2060.txt>, 2004 - 04 - 08
- [4] Borenstein N, Freed N. Multipurpose internet mail extensions[DB/OL]. <http://www.isi.edu/in-notes/rfc1341.txt>, 2004 - 04 - 20