

# 基于MQ的文件分片传输系统的设计与实现

吴成宾, 朱 彬

(成都大学信息网络中心 成都 610106)

**【摘要】**基于MQ的文件分片传输系统融合了通信中间件、Zlib压缩、对称加密等技术,能够实现异构环境下可靠的文件数据交换。该文研究分片文件系统的主要组成部分,围绕系统发送和接收两方面的功能作了详细的阐述,并讨论了文件分割与还原、加解密等算法。

**关键词** 解密; 加密; MQ通信中间件; 文件分片; 压缩文件库  
中图分类号 TN919.3 文献标识码 A

## A Splitting Files Transmission System Based on MQ

WU Cheng-bin, ZHU Bin

(Network and Information Center, Chengdu University Chengdu 610106)

**Abstract** A splitting files transmission system based on Message Queue (MQ) is integrated with the technologies of middleware, zlib, and symmetric key cryptography. Such a system can achieve secure file exchange in heterogeneous environment. This paper introduces the major components of the system, The sending and receiving subsystems are discussed in detail. The splitting and recovering, encryption, and decryption of the algorithm are investigated.

**Key words** decryption; encryption; MQ communication middleware; file slice; file compressing Zlib

在电子政务、电子化金融交易等许多重要环境中,都需要进行可靠的文件数据交换,采用传统的通讯手段(比如HTTP、FTP、E-Mail等),可靠性、安全性得不到充分保证。

消息队列(Message Queue, MQ)是IBM公司于20世纪80年代推出的一款消息中间件,目的是在任意两个分布式进程之间进行可靠的数据传输。MQ支持异步数据传输,并支持多种底层通信协议,如TCP/IP。

消息队列有许多显著的优点,可在不同的平台上使用相同的应用程序接口,实现跨平台通信,对消息的创建和发送不依赖于时间,增加了处理的灵活性。

### 1 异构环境下的文件分片传输系统

利用MQ的性能特点和高可靠性,本文设计了一个异构环境下的文件分片传输系统,提供了一种可靠的文件传输手段。该系统由发送方、接收方软硬件、MQ进程、MQ通信接口、本地文件系统与数据源,以及网络等组成,具有文件收发两方面的功能。

收发双方进程之间交换的基本数据单位是一个个切分好的文件分片,各分片由字段组成,各字段之间用0x02来分隔。分片报文SLICEPACKET结构如表1所示。

表1 报文格式结构

字段	类型与长度	说明
BranchID	Char(4)	机构代码
SerialNo	Char(8)	序列号,用于区分新老数据
FileName	Varchar(255)	文件名,变长
TotalSlice	Int(4)	总片数,无符号整数,转换成整数串形式(下同)
SliceNo	Int(4)	当前分片编号,无符号整数
SliceLen	Int(4)	当前分片长度,无符号整数
SliceContent	Blob	分片内容

#### 1.1 MQ通信接口类

MQ通信接口类提供打开MQ队列、往MQ队列置入数据、从MQ队列摘取或浏览数据等接口。收发双方的应用进程都通过该接口与MQ进程交换数据。下面给出该类的主要结构<sup>[1]</sup>。

```
#include <imqi.hpp> //MQ调用接口头文件  
class CMQComm //MQ通信接口类  
{
```

收稿日期:2006-04-30

作者简介:吴成宾(1975-),男,硕士,高级工程师,主要从事中间件、计算机网络与技术方面的研究;朱彬(1961-),女,副教授,主要从事信息与通信系统方面的研究。

```

    BOOL InitQMgrConnect(LPCTSTR
szQMgrName); //初始化队列管理器
    CMQComm(LPCTSTR szQMgrName,int
iSendQueueNum,LPCTSTR
szSendQueueName,BOOL bMultiQueue,LPCTSTR
szRcvQueueName,LPCTSTR szDestCode);
    int MQSendMsg(LPCTSTR szDestCode,LPTSTR
szbuf,int iSendLen,int iPriority=0);
    int MQRcvMsg(int iQueueID,LPTSTR
szbuf,DWORD dLen);
    int MQOpenQueue();
    void MQDisconnect();
    ...
};

```

## 1.2 发送方软件

发送方软件主要完成文件的压缩、加密和分割功能。分割算法为：

(1) 设定标准分片的大小,约定发送方和接收方软件都使用这个标准值,标准分片大小为64 KB:  
const int SLICESIZE = 64 × 1 024。

(2) 取得待发文件长度FileLen,计算最大可分割片数,TotalSlice = 「FileLen / SLICESIZE」,符号“「」”表示向上取整。显然,除了最后一个分片可能不足标准分片长度之外,其余分片的大小都是一致的。

(3) 计算最后一个分片的大小,LastSliceSize = FileLen - (TotalSlice - 1) × SLICESIZE,当文件长度不足SLICESIZE时,LastSliceSize = FileLen。

(4) 打开文件,从头开始顺序读取第 $n(n = 0,1,2, \dots, \text{TotalSlice}-1)$ 个分片。当 $n > 0$ 且 $n \neq (\text{TotalSlice}-1)$ 时,读取长度为SLICESIZE,否则为LastSliceSize。

(5) 将读取的分片数据打成SLICEPACKET格式的报文,并附加上其它的包头信息,再通过MQ通信接口将数据包丢到MQ发送队列。

由于大批量数据的加密、解密,本文采用把公共密钥与对称密钥技术相结合的折衷方案<sup>[2]</sup>,即在双方正式传输文件前,用公共密钥技术传送对称密钥,然后再用对称密钥对实际传输的文件数据加密并对不同的分支机构分配不同的密钥,从而有效地提高了安全性。

## 1.3 接收方软件

接收方软件的任务是收集各发送方传来的文件分片,进行重组及解压缩、解密,然后将得到的原始数据文件交业务线程进行后续处理。接收方主函

数工作流程为：

(1) 从MQ队列摘取一个数据报文,从中提取出分片数据。

(2) 取得机构代码和本次的序列号(发送批次)。

(3) 取得文件名、分片总数、当前分片编号和当前分片长度。

(4) 读取当前分片内容。

(5) 调用写分片文件函数。

(6) 登记或重组各文件分片,并判断某文件的各分片是否已收齐。

写分片文件函数的方法为：

(1) 当收到某个文件的第一个分片(不一定是分片编号为0的那个分片)时,即创建该文件,并确定文件长度。

(2) 取得分片报文中的TotalSlice-1(当TotalSlice=1时)及当前分片长度时,创建一个FileLen长的文件,内容任意,再把第一个分片的内容拷入文件中。

(3) 以后每次取得该文件的一个分片时,计算修改文件时的偏移量和修改长度：

Offset = 当前分片编号 × 当前分片长度

ModifyLen = 当前分片长度

用这种方法,即使分片在传输的过程中被打乱了次序,也可以正确实现重组<sup>[3]</sup>。

## 2 数据格式转换与容错

发送方进程传输的数据源自数据库时,必须保留数据类型等格式信息,以保证接收方能正确处理并达到精度要求。发送方进程按照下发的检索条件从SQL数据源中提取带格式的原始数据信息,根据SQL表的字段信息生成相应的Dbf文件,然后将记录数据一一对应地写入Dbf中,最后经压缩、加密后逐片发送出去。接收方进程还原Dbf文件后,逐条读取记录写入本地中心数据库。

SQL数据库和Dbf文件主要字段类型的对应关系如表2所示<sup>[4]</sup>。

表2 SQL和Dbf文件主要字段类型对照表

SQL字段	Dbf字段	说明
char	Charater	长度保持不变
varchar	Charater	长度保持不变
decimal	Numeric	长度保持不变
int	Numeric(16,0)	长度16 B
smallint	Numeric(8,0)	长度8 B
tinyint	Numeric(4,0)	长度4 B
datetime	Charater(21)	长度21 B 格式为“YYYY-MM-DD hh:mm:ss.xxx”

关于数据容错,分片结构中有一个字段SliceNo表示批次,当接收线程收齐某个文件的所有分片之后,在解压缩之前,检测系统中有无业务线程正在

处理一个批次号小于当前接收批次的同一个发送进程的同一个文件,若有,则需要给该线程置上“脏”数据标志,否则就解压缩文件:

```
EnterCriticalSection;//进入临界区
    m_bDirtyData = TRUE;//置脏数据标志
    ...
LeaveCriticalSection;//退出临界区
try
{
调用Zlib库例程解压缩...
    解密...
}
catch(...)
{
异常处理...
}
```

当业务线程检测到“脏”数据标志时,中止处理该文件,并重新初始化相关的环境后退出。

MQ数据队列就好像一个数据仓库<sup>[5]</sup>,只要磁盘空间足够,仓库中的内容既不会丢失,也不会改变。所以只要在应用进程终止前,将描述分片的结构内容写入磁盘,进程下次启动时,从磁盘中读入保存的内容,然后打开MQ队列摘取报文,就可实现断点续传,这对于挽救大文件非常有用。

MQ队列在重负荷情况下不够稳定,常造成MQ系统崩溃。应用进程必须确保首先摘除队列上的报文并还原成文件以备后续处理,如果还原后的文件过多,业务线程来不及处理,此时需要将未处理的文件名及其路径、文件来源、发送方机构代码等信息写入待处理文件队列,一旦有业务线程空闲下来,就按照FIFO原则处理该队列中的文件。

表3 新旧系统测试结果对比表

对比项	旧系统	基于MQ的新系统
是否加密	否	采用对称加密算法DES进行大批量数据的加密
是否压缩	否	采用zlib压缩库
文件压缩率	不压缩	压缩到原来的10%~16%
是否支持容错	否	支持,能防止旧数据覆盖新数据,等等
是否支持文件断点续传	否	支持,并且关机后也能断点续传
MQ接收队列负荷及平均报文数	负荷重,报文数常常 2 500	轻,报文数常常 1 000
单台服务器每8 h能处理的业务数据量	500 MB	2.5~4 GB
数据中心是否支持对分支机构数据的按任意条件、任意时段的选择性提取	不支持	支持,通过下发检索条件,可控制客户端进程按要求提取当前和历史数据库中的记录,并生成文件压缩加密后上传

从表中可知,新系统在处理能力、可靠性、安全性、灵活性等方面都比旧系统有较大的改进。

### 3 结束语

本文对文件分片系统进行了论述,其中重点是分割、还原算法、密钥方案的选择,以及异构数据库之间的数据类型转换、“脏”数据、断点续传等容错方面的解决方案。系统具有实用性、安全性、可靠性和实时性,可以应用在所有需要确保可靠的文件数据交换的场合。

### 参考文献

- [1] JAMALEDDINE B W. IBM websphere应用服务器程序设计[M]. 朱毓斌,吴飞,译.北京:清华大学出版社,2003.
- [2] 帅红涛,舒勤,吴戈. MQ series 在高速公路联网收费系统中的应用[J]. 计算机应用,2001,4: 100-101.
- [3] 游晗,和应民. 通信中间件在GSM网络监控系统中的应用[J]. 自动化技术与应用,2005,05: 39-40.
- [4] 胡欣. MQSeries队列管理器的远程管理[J]. 铁路计算机应用,2003,01: 20-23.
- [5] 张鹤颖,龚文华. 主动队列管理机制的性能分析[J]. 计算机科学,2004,04: 80-83.

编辑 熊思亮