

# 一种商用文件传输系统<sup>①</sup>

王佳亮

(上海浦东发展银行信用卡中心, 上海 200120)

**摘要:** 研究了一种在商业上使用的高可靠性和高效性的文件传输系统. 基于 Socket 传输, 通过 XML 对传输信息进行记录与信息交互, 对传输系统进行了完善的传输保障机制设计, 在传输过程中发生的网络瞬断, 在网络恢复时, 系统可以自动恢复传输, 并可以对于一些已经传输且未传输完成的文件进行断点续传. 系统具有极好的容错机制, 不会由于一些传输异常而导致系统崩溃. 最后, 商用文件传输系统有详细全面的日志记录, 万一当系统发生问题时, 可以准确定位问题, 解决问题. 系统配置简单, 操作方便, 在传输大文件时, 由于支持断点续传, 因此保证了传输的效率, 并在 Windows 7 企业版系统中测试通过.

**关键词:** socket; 文件传输; 断点续传

## Commercial File Transmission System

WANG Jia-Liang

(Shanghai Pudong Development Bank Credit Card Center, Shanghai 200120, China)

**Abstract:** Studied on a commercial use of high reliability and efficiency of file transfer system. Based on the Socket transmission. Records and information of transmission through XML information interaction, Perfect transmission guarantee mechanism for transmission system design, when it transfers the file, the network transient fault, then the network recovery, the system can automatically recover the transmission, and it can transfer the file which not completed transmission. The system has the good fault tolerance; it cannot breakdown because of the abnormal transmission. The last, the commercial file transmission system has the detailed logging in an all-round way, when the system has the problem, it can position the problem accurately, and solves the problem. System configuration is simple, and easy to operate, when transfer the large file, supporting resume transmission, so it ensures the efficiency of transmission. The system test passed in Windows 7 Enterprise Edition system.

**Key words:** socket; file transfer; breakpoint resumes transmission

Internet 从 1969 年投入使用以来, 到如今, 已有 40 多年的时间. Internet 的出现极大地方便了人们的工作和生活, Internet 最常用的应用层协议, 很多人已不陌生, 如: Telnet、FTP、HTTP、SMTP、DNS 等. Internet 的 FTP 协议应用很广泛, 市面上基于 FTP 的传输软件也很多, 单仅仅是作为基本的文件传输使用, 这些软件功能简单, 当使用环境对数据传输要求比较高且复杂时, 这些软件往往力不从心. 常见的客户端文件传输软件, 如: CuteFTP<sup>[1]</sup>, LeapFTP<sup>[2]</sup>与 FlashFXP<sup>[3]</sup>. 在 FTP 服务器端软件中, Serv-U<sup>[4]</sup>是一种被广泛运用的

FTP 服务器端软件, 支持 3x/9x/ME/NT/2K 等全 Windows 系列. 可以设定多个 FTP 服务器、限定登录用户的权限、登录主目录及空间大小等, 功能非常完备. 它具有非常完备的安全特性, 支持 SSI FTP 传输, 支持在多个 Serv-U 和 FTP 客户端通过 SSL 加密连接保护您的数据安全等. 以上这些, 配合起来使用, 可以实现文件的传输的功能, 但是, 用于商业, 缺乏一个统一的管理平台, 实现文件传输的自动化也比较困难, 也不支持文件的并行传输. 文献[5]提出了并行下载最优机制, 从而优化对等(peer-to-peer, 简称 P2P)系

<sup>①</sup> 收稿时间:2013-08-08;收到修改稿时间:2013-08-26

统的并行下载. 在日常使用中, 由于对等网络对一些文件往往可以提供数百个源节点, 对于一些热门的资源, 往往可以达到上千, 并行下载可以获得良好的效果. 但是, 在商用环境中, 使用环境相对封闭, 源节点往往就只有一个, 效果就不会很好.

商业上的文件传输, 一般在企业的局域网内对相关文件进行传输, 首先必须要满足高可用性的要求, 有完善的异常处理机制, 当传输大文件时, 可以实现断点续传, 同时, 商用文件传输, 也要满足一定的传输时效要求, 传输效率要高. 传输系统对文件源传输的直接好处是传输的总带宽增加了, 带宽利用率得到提升, 当源文件的节点突然离开网络或者发生故障时, 传输系统也能提供更好的稳健性. 对于大文件传输, 效果尤为明显, 发生传输错误时, 只需将未传完的部分传输即可, 不必整个文件重新传输, 传输的效率也大大提高<sup>[6]</sup>.

本文所设计的商用文件传输系统使用环境的前提是企业内部网络是安全的, 然后本文从商用角度出发, 结合商用实际使用环境, 通过对文件传输服务器以及客户端统一进行系统分析, 设计和软件架构, 达到商用文件传输使用的目的.

## 1 商用文件传输系统需求分析

### 1.1 用户需求

商用文件传输系统, 主要用于企业内部网络中文件的传输. 首先, 系统间的文件传输有很多, 不同的系统间文件传输, 又需要有不同的权限. 同时, 各个系统间文件传送的路径也不同. 因此, 商用文件传输系统, 需要具备完善的权限机制, 和传输分配机制, 不同的权限配置用户进行文件传输, 系统可以按照权限与传输配置, 自动分配传输路径, 达到文件自动传输的目的. 其次, 商用文件传输系统有完善的传输保障机制, 在传输过程中发生的网络瞬断, 在网络恢复时, 系统可以自动恢复传输, 并可以对于一些已经传输且未传输完成的文件进行断点续传. 系统应该具有极好的容错机制, 不应该由于一些传输异常而导致系统崩溃. 最后, 商用文件传输系统应该有详细全面的日志记录, 万一当系统发生问题时, 可以准确定位问题, 解决问题.

### 1.2 分析建模

根据商用文件传输系统实际的用户需求, 结合系

统的实际运行情况, 构建出系统相关功能模块. 系统用户管理配置模块, 用于对用户权限和文件传输规则进行配置, 如用户注册, 用户登录验证, 用户文件传输路径配置, 用户文件传输时间配置等. 系统管理模块, 用于对系统相关的参数进行配置, 如服务端地址和端口配置, 日志路径地址配置, 断点续传间隔时间配置等. 文件传输模块, 主要负责服务端和客户端的文件传输, 包括正常客户端与服务端的文件传输, 以及对客户端与服务端传输异常的文件进行续传. 系统日志模块, 主要负责对系统运行中的状况进行记录, 生成日志并可以对相关日志进行调阅. 本文使用 Microsoft Visio 2010 进行系统相关图的设计, 商用文件传输系统的功能层次模型图如图 1 所示.



图 1 商用文件传输系统功能层次模型图

商用文件传输的实现, 主要需要解决三个方面的问题: (1)极强的文件传输容错机制, 在网络发生瞬断, 对文件已传输的部分不会丢失, 当网络恢复时, 系统可以自发继续传输; (2)功能完善的权限与传输配置; (3)详细全面的日志记录.

## 2 商用文件传输系统概要设计

考虑到商用文件传输系统数据交互的特点, 因此系统采用 C/S 结构设计, 并以星型拓扑作为客户端与

服务端的连接形式. 商用文件传输系统, 主要分为两个部分, 一个是服务端软件, 一个是客户端软件. 系统以服务端为主要核心节点, 服务端程序部署在服务器上, 提供数据接收与分发的服务, 客户端软件部署在客户机上, 用于向服务端发送文件. 对于数据传送异常的处理原则是客户端负责传输失败后重新发起传输请求. 商用文件传输系统的总体传输流程如下:

(1) 在服务端软件配置传输的用户名和密码, 以及此用户与服务端接收的目录, 同时设置服务端软件所在机器的 IP 和端口, 日志保存路径.

(2) 客户端与服务端连接成功后, 用户选择需要上传的文件上传, 客户端记录所传输的文件名称和文件路径, 服务端记录客户端通过 Socket 发过来的文件信息<sup>[7]</sup>, 并反馈客户端可以传输, 客户端开始传输文件. 服务端接收.

(3) 若文件传输成功, 服务端反馈客户端传输完成, 传输结束. 并将各自的传输记录文件移入传输已完成的文件夹内.

(4) 若文件传输失败, 如网络断开, 传输中断, 客户端会按照配置的时间, 发起与服务器的连接请求, 当客户端与服务端再次连接成功后, 客户端通过检索传输记录, 得到传输失败的文件, 发送给服务器, 服务器检索已经传输的文件大小, 反馈给客户端, 客户端进行断点续传. 传输成功后, 服务端反馈客户端传输完成, 传输结束. 并将各自的传输记录文件移入传输已完成的文件夹内.

(5) 客户端与服务端的连接, 文件传输, 系统都会对这些行为进行记录, 以日期为文件名生成系统日志记录文件.

系统的业务流程图, 如图 2 所示.

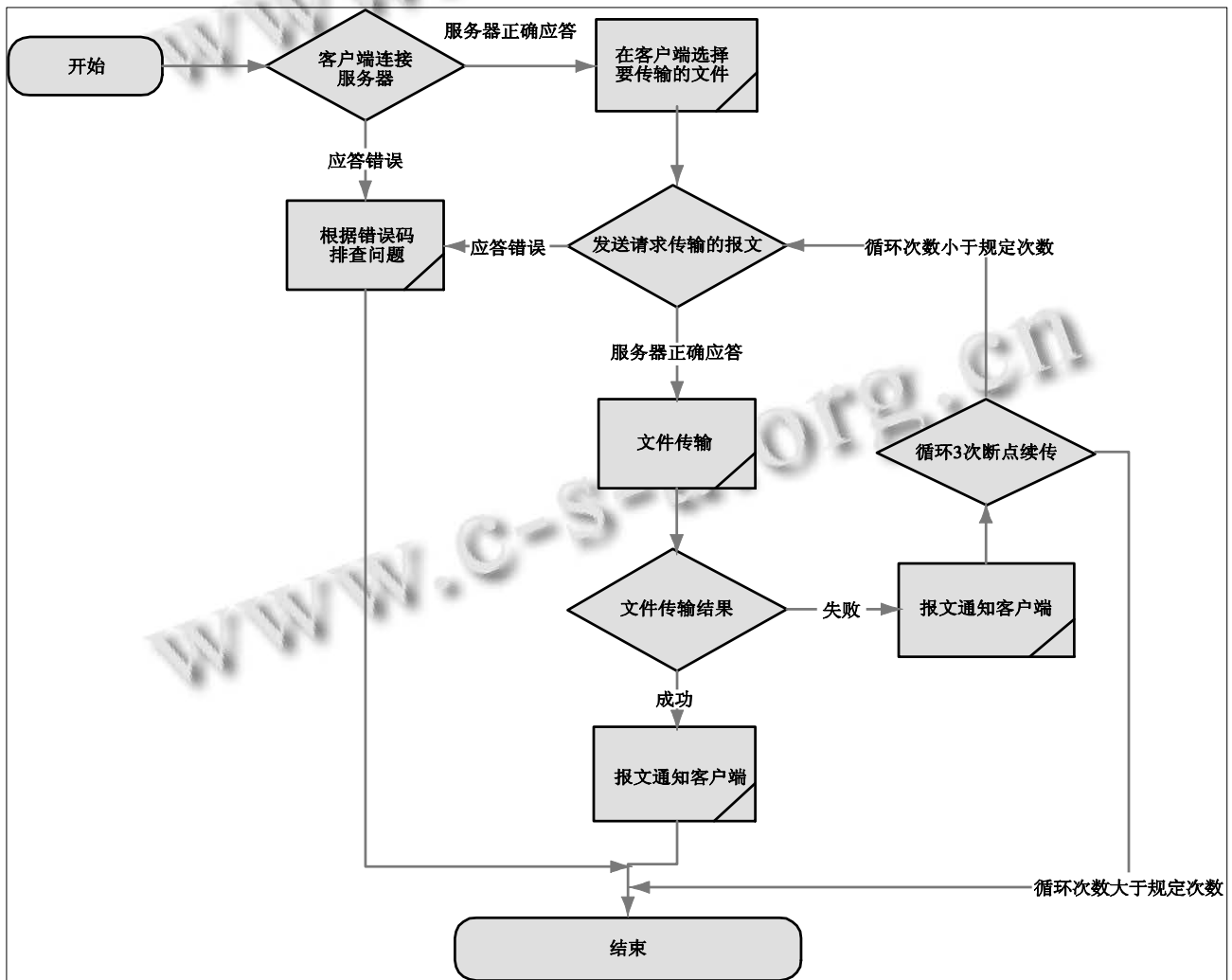


图 2 商用文件传输系统业务流程图

### 3 商用文件传输系统传输报文设计

#### 3.1 报文定义

报文主要分为报头, 报文体和报文尾三个部分, 报头由三位代码组成, 客户端发送的以字母 C 开头, 服务端发送的由 S 开头. 报文体包含多个信息, 包含多个信息间用字符"!@#!" 分割, 报文尾统一由字符"\r\n"结尾. 报文格式如下所示.

报头( 3 位)	报文体	报文尾
----------	-----	-----

#### 3.2 客户端、服务端报文设计

(1) 客户端与服务端请求连接

C01	用户名!@#! 密码	\r\n
-----	------------	------

(2) 服务端对客户端请求连接的响应

S01	连接成功, 可以传输	\r\n
S02	连接失败 - 用户名密码错误	\r\n

(3) 服务端向客户端传输文件

C11	用户名!@#!密码!@#!文件GUID 编号!@#! 文件名	\r\n
-----	-----------------------------------	------

(4) 服务端对客户端的传输响应

S11	服务器已登记将要传输的文件信息, 客户端可以传输	\r\n
S12	连接失败-用户名密码错误	\r\n
S13	服务器登记将要传输的文件信息失败	\r\n

(5) 客户端文件传输完成服务端的反馈

S21	文件传输成功!@#!文件GUID编号!@#! 文件名	\r\n
-----	-------------------------------	------

(6) 客户端请求关闭连接

C21	用户名!@#!密码!@#!	\r\n
-----	---------------	------

(7) 客户端向服务端发送断点续传请求

C31	用户名!@#!密码!@#!文件GUID编号	\r\n
-----	-----------------------	------

(8) 服务端向客户端响应断点续传请求

S31	验证成功!@#!文件已传输的字节大小	\r\n
S32	连接失败-用户名密码错误	\r\n
S33	服务端没有需要断点续传的文件信息	\r\n

### 4 商用文件传输系统详细设计

#### 4.1 XML 操作模块

商用文件传输系统中, 用户名, 密码的配置, 用户上传路径的配置, 客户端传输的记录文件, 服务端传输的记录文件, 都在相关的 XML 文件中进行数据交互. XML 操作模块, 主要对相关的 XML 文件进行创建<sup>[8]</sup>, 读取以及对其内容进行编辑或删除.

(1) 用户名, 密码, 文件上传保存路径信息 XML 格式:

```
<?xml version="1.0" encoding="utf-8"?>
<UserConfig>
  <UserInfo UserName="用户名" UserPassword="
  密码">
    <FilePath>服务器保存传输文件的路径
  </FilePath>
  </UserInfo >
</ UserConfig >
```

(2) 客户端记录文件 XML 格式:

```
<?xml version="1.0" encoding="utf-8"?>
<ClientTransfer>
  <FileInfo FileName="要记录的文件名" Guid="
  为所传输文件分配的 Guid">
    <FilePath>要传输的文件全路径</FilePath>
  </FileInfo>
</ClientTransfer>
```

(3) 客户端记录文件 XML 格式:

```
<?xml version="1.0" encoding="utf-8" ?>
<ServerTransfer>
  <FileInfo FileName="服务器记录的文件名"
  Guid="传输文件的 Guid">
    <FileServerPath>服务器保存文件的全路
  </FileServerPath>
  </FileInfo>
</ServerTransfer>
```

#### 4.2 前端文件传输模块

客户端首先输入用户名和密码, 向服务端发送 C01 报文请求与服务端连接. 若连接成功, 且用户名和密码正确, 服务端向客户端返回 S01 报文, 否则返回 S02 报文<sup>[9]</sup>. 客户端接收到服务端的 S01 报文后, 用户从客户端向服务端发送文件, 客户端会为即将发送文件, 分配一个 Guid 编号, 客户端会对即将发送的文件

进行 XML 文件记录, 生产的 XML 文件以传输文件的 Guid 编号为文件名, 保存在客户端的未完成传输任务目录下, 然后客户端向服务端发送含有用户信息和文件信息的 C11 报文, 服务端接收到后, 会对服务端发来的文件信息进行 XML 记录, 同样文件名采用从客户端传来的 Guid 来进行命名并保存在服务端的未完成传输任务目录下. 服务端记录成功后, 向客户端返回 S11 报文. 客户端向服务端传输文件. 服务端文件接收完成后, 向客户端发送 S21 报文后关闭连接. 客户端和服务端的文件记录 XML 文件, 从未完成传输任务目录移入到已完成传输任务目录下. 文件传输结束.

### 4.3 后端断点续传模块

客户端会按照设定的时间间隔, 不断循环读取未完成传输任务目录下的 XML 文件, 根据 XML 文件中记录的内容, 向服务端发送 C31 报文, 服务端接收到信息后, 获得客户端报文中的 Guid 值, 检索服务端未完成传输任务目录下是否有此文件, 有的话根据服务端记录的文件路径信息, 获得未完成传输的文件, 并

获得未完成传输的文件大小, 向客户端返回 S31 报文, 客户端获得未传输文件的大小后, 进行运算后, 对未完成部分进行续传. 如果服务端没有客户端发来的 C31 报文中所请求的文件, 则返回 S31. 客户端 XML 文件记录则移入到异常传输记录文件夹, 用于手工处理. 文件传输成功, 客户端和服务端的文件记录 XML 文件, 从未完成传输任务目录移入到已完成传输任务目录下. 文件断点续传结束.

### 4.4 日志模块

客户端日志, 和服务端日志, 都以天为单位记录. 日志信息包括时间, 传输客户端的 IP, 客户端的用户名, 传输的文件时系统所分配的 Guid 编号, 文件名, 和文件传输路径, 传输结果. 日志记录格式如下:

记录时间(年-月-日 时:分:秒)	记录信息	\r\n
----------------------	------	------

### 4.5 UML 建模

将系统的详细设计, 进行 UML 建模<sup>[10]</sup>, 生成的设计类图, 如图 3 所示.

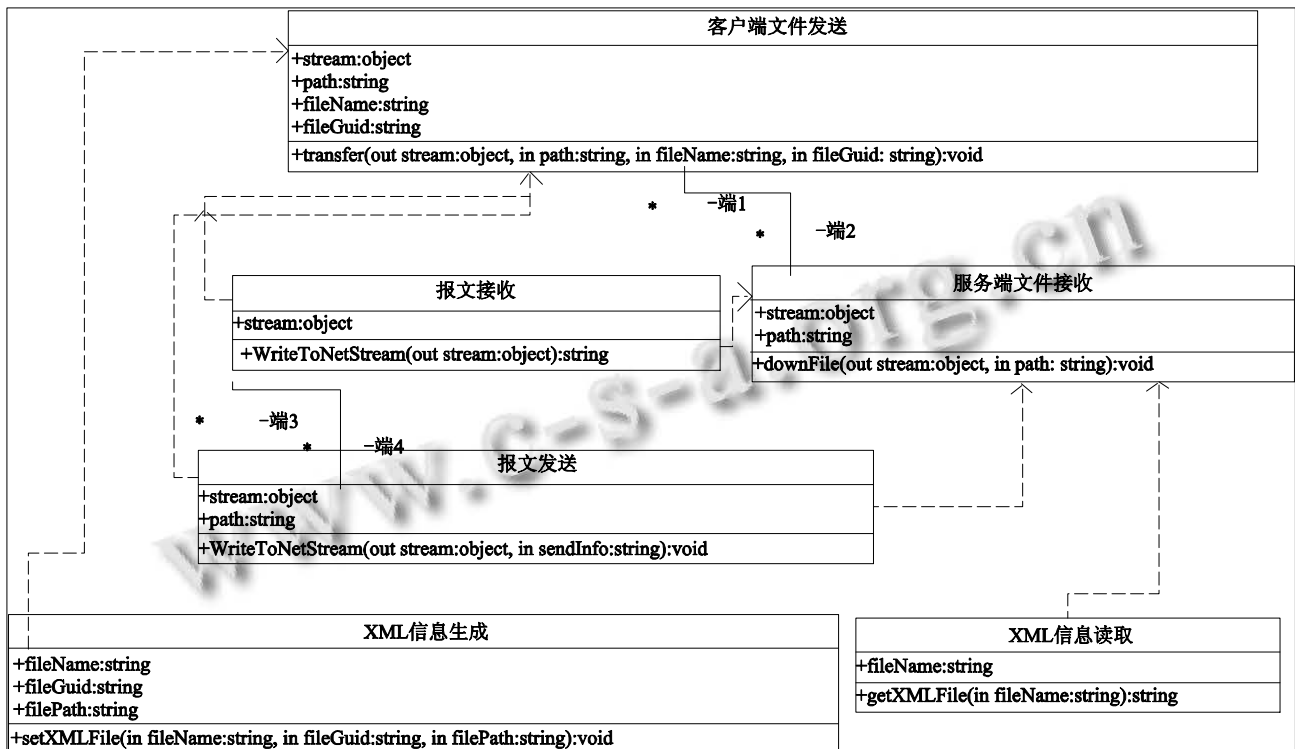


图 3 商用文件传输系统设计类图

## 5 商用文件传输系统的实现

系统客户端上传文件选取了一个 58M 的 PDF 格式的文件, 系统传输正确. 同时, 模拟了系统文件传

输过程中由异常引发的传输中断, 在系统恢复后, 文件断点续传功能实现, 文件传输正确. 达到了最初的设计要求. 实现过程截图, 商用文件传输系统的客户

端界面如图 4 所示, 商用文件传输系统的服务端界面 如图 5 所示.

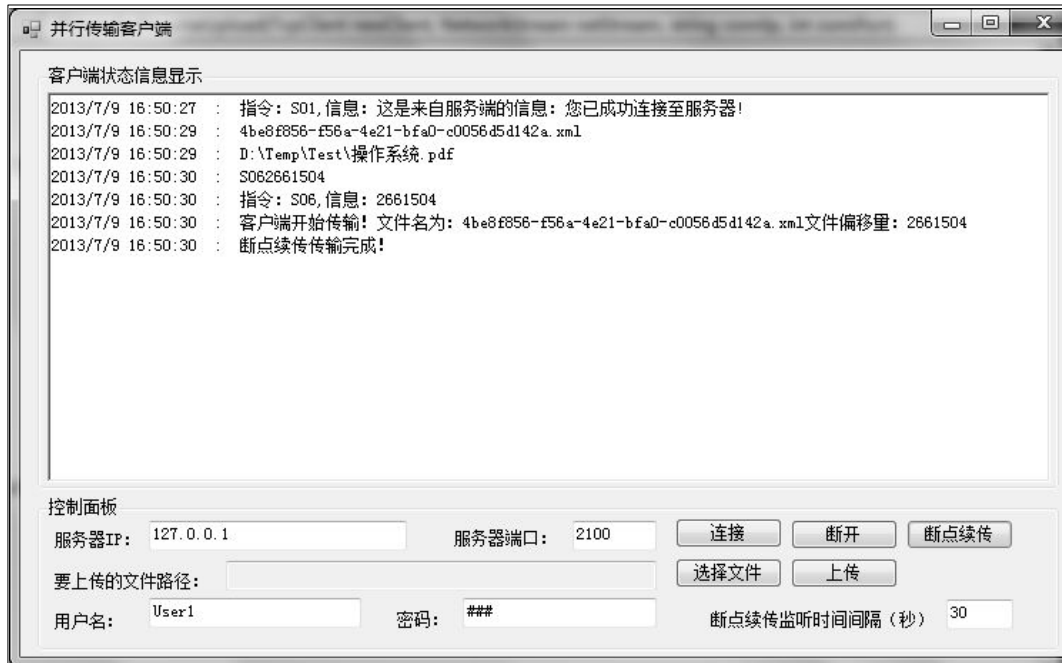


图 4 商用文件传输系统的客户端

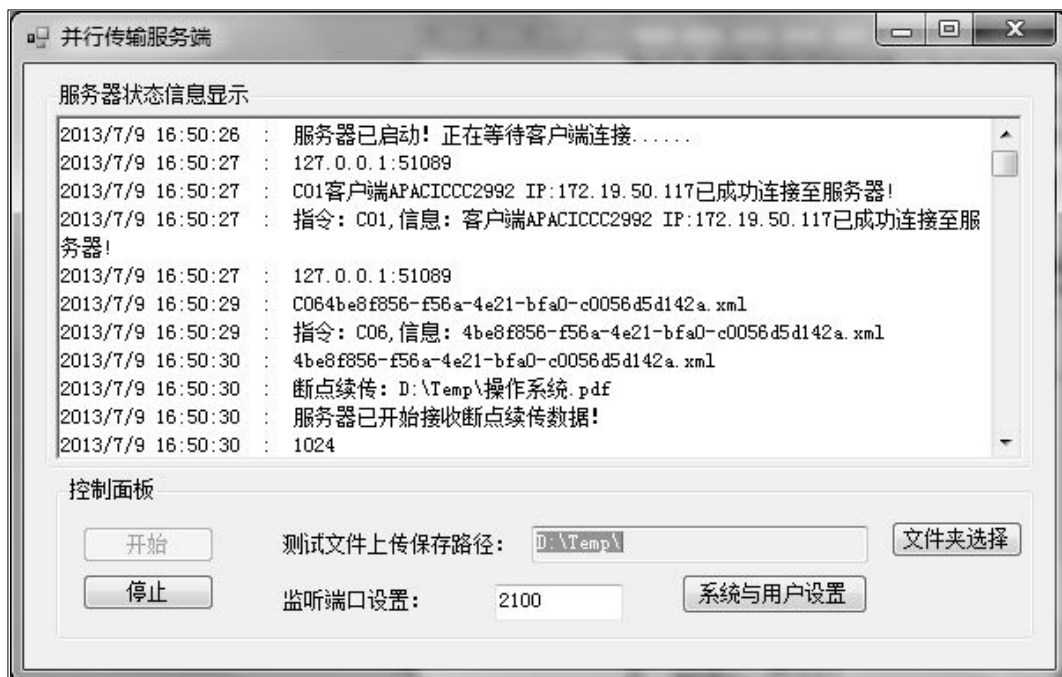


图 5 商用文件传输系统的服务端

## 6 结语

本文讨论的是基于 Socket 的商用文件传输系统的设计与实现. 满足了最初的商用传输的设计要求, 实现了商用文件传输的功能, 系统配置简单, 操作方便,

在传输大文件时, 由于支持断点续传, 因此保证了传输的效率. 并在 Windows7 企业版系统中测试通过. 文件配置和信息记录采用 XML 数据格式, 简化了配置, 并提升了效率. 本系统主要有如下特点:

- (1) 详细的信息反馈机制.
- (2) 可靠的传输保证机制(断点续传).
- (3) 操作简单, 便捷.
- (4) 完善的日志记录, 易于排错.

由于目前大多数企业的文件传输使用的是操作系统自带的FTP功能, 配置复杂, FTP较为分散且不利于管理, 也没有断点续传功能. 由于大多数企业数据还没有达到一定规模, 商业的文件传输系统并没有得到企业的重视, 商用文件传输系统还没有实际应用的环境, 但随着企业不断发展, 数据量不断增大, 大数据时代的逐渐来临, 商业文件传输, 作为客户端与服务端的数据传输工具, 极大地方便了工作, 提升了工作效率. 如果在此基础上对系统进行进一步完善, 构建复杂大型的文件传输引擎, 将会产生极大的商业价值和经济效益.

#### 参考文献

- 1 百度百科 .CuteFTP 介绍 .http://baike.baidu.com/view/177879.htm.2013-3-13.

- 2 百度百科 .LeapFTP 介绍 .http://baike.baidu.com/view/119580.htm.2013-3-13.
- 3 百度百科 .FlashFXP 介绍 .http://baike.baidu.com/view/177890.htm.2013-3-13.
- 4 百度百科 .Serv-U 介绍 .http://baike.baidu.com/view/537933.htm.2013-3-13.
- 5 王正, 罗万明, 阎保平. 并行下载最优机制. 软件学报, 2009, 20(8).
- 6 邓湘, 吴迪. 基于 P2P 的文件并行上传机制研究. http://www.doc88.com/p-995532369499.html. 2013-3-15.
- 7 志良的技术博客. 深入探析 c# Socket. http://www.cnblogs.com/tianzhiliang/archive/2010/09/08/1821623.html. 2013-3-15.
- 8 Palmer G. 康博译. C#程序员参考手册. 北京: 清华大学出版社, 2002.
- 9 周存杰. Visual C#.NET 网络核心编程. 北京: 清华大学出版社, 2002.
- 10 曾强聪. 软件工程. 北京: 高等教育出版社, 2004.

(上接第 107 页)

集, 聚类起来相对容易, 均能获得较好的聚类效果. 但对比 Aggregation1、Aggregation2、Aggregation3 这三个较难的数据集来说粒子群算法的聚类精度就明显不如 PSO-AP 算法. 总的来说 PSO-AP 算法在对数据集的适应度与聚类精度上都要优于粒子群算法与普通的 AP 算法.

#### 5 结语

本文针对近邻传播(AP)算法中偏向参数与收敛系数对 AP 算法的聚类效果的局限性的问题进行了分析研究. 首先, 本文通过实验分别对偏向参数与收敛系数进行调试取值, 研究了偏向参数与收敛系数的特性. 然后, 结合偏向参数与收敛系数的特性, 提出了通过粒子群算法搜寻最优的偏向参数与收敛系数的方法, 使近邻传播算法达到一个最佳的聚类结果. 最后, 通过实验证明了 PSO-AP 算法的有效性, 且在聚类精度与适应性上要优于普通的 AP 算法与粒子群算法.

#### 参考文献

- 1 Frey BJ, Dueck D. Clustering by passing messages between

data points. Science, 2007, 315: 972-976.

- 2 王开军, 张军英等. 自适应仿射传播聚类. 自动化学报, 2007, 33(12): 1242-1246.
- 3 刘胜宇, 刘家锋等. 基于改进 AP 聚类算法的人脸标注技术研究. 智能计算机与应用学报, 2011, 1(1): 35-38.
- 4 杨传慧, 吉根林等. AP 算法在图像聚类中的应用研究. 计算机与数字工程学报, 2012, 40(10): 119-121.
- 5 周世斌, 徐振源等. 一种基于近邻传播算法的最佳聚类数确定方法. 控制与决策学报, 2011, 26(8): 1147-1157.
- 6 肖宇, 于剑. 基于近邻传播算法的半监督聚类. 软件学报, 2008, 19(11): 2803-2813.
- 7 邢艳, 周勇. 基于互近邻一致性的近邻传播算法. 计算机应用技术, 2012, 29(7): 2524-2526.
- 8 刘靖明, 韩丽川等. 基于粒子群的 K 均值聚类算法. 系统工程理论与实践, 2005, 6: 54-58.
- 9 陶新民, 徐晶等. 一种改进的粒子群和 K 均值混合聚类算法. 电子与信息学报, 2010, 32(1): 54-58.