基于HTTP协议和数据库的文件上传方法*

Method of Uploading Files Based on Database and HTTP Protocol

黄景文

Huang Jingwen

(广西大学信息网络中心,广西南宁 530004)

(Information & Network Center, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:提出一种基于HTTP协议和数据库的文件上传方法,给出相应的实现算法。该方法通过表单提交文件信息,从多域数据流析取文件内容,并以二进制方式存入数据库的BLOB字段,从而实现文件上传。该方法具有结构化信息和非结构化信息同步上传、长文本和带格式文本可以以文件形式上传到数据库、维护难度低、扩展和移植方便等特点。

关键词:文件 数据库 HTTP协议 上传 多域数据流 析取

中图法分类号:TP393.03;TP311.13 **文献标识码**:A **文章编号**:1002-7378(2005)03-0186-03

Abstract: A method of uploading files based on database and HTTP protocol is developed. In this solution, the file information is sent in the browser side via filling of HTML form, and extracted from multipart stream in the server side, and stored as binary stream into the BLOB field of database. In this method, the structured information and unstructured information can be submitted synchronously, and large text and formatted text can be uploaded to database as a file. The data maintenance and transplant between different system platforms are easy.

Key words: files, database, HTTP protocol, uploading, multipart stream, extract

文件上传是网络系统最活跃的应用之一。文件上传一般使用FTP或HTTP协议。前者本质上是异步的,需要借助于FTP客户端软件,其缺点是可编程性较差,不便于B/S架构下特定应用的集成。HTTP上传可产生许多可访问的附加信息,具有在服务器端的可编程性和用户端的直接性,以及操作简单、登录快速、交互性强等特点,而且能够使用SSL编码对上传信息进行加密,使得HTTP上传比FTP上传有明显的优势[1]。在处理基于B/S的文件上传时通常使用HTTP协议。

无论是FTP还是HTTP,传统的上传方法必须依赖于文件系统,需要为上传者开放服务器端的文件系统的读写权限,存在一定的不安全因素,而且数据存储分散、数据冗余、不一致,易被修改,可靠性差,应用于量大的用户资料上传管理时问题尤为突出。本文提出一种使用HTTP协议的基于数据库的文件上传方法,从多域数据流析取文件内容,并以二进制方式存入数据库的BLOB字段,实现文件上传。

1 基于HTTP的文件上传分析

RFC1867[2]规定了使用浏览器的文件上传规则,前端利用HTML页面的Form表单和Post方法以Multipart/form-data编码方式向服务器提交多域数据流,Action属性指定的后端程序对截获的多域数据流解码获得上传内容并存入磁盘文件或数据库。因此,获知多域数据流的格式是完成解码程序的基础工作。

本文采用ASP技术将截获的数据流以Response.BinaryWrite写回浏览器,并查看生成的HTML文件的源代码的方法研究上传的数据流。实验表明,通过浏览器提交的多域数据流(仅为文件域片断)大致格式如图1所示。

根据RFC1867规范[2],多域数据流由多个数据段组成,每段来自一个表单域,且位于边界分割符(如图1的"————7d42802160348",由多域数据

收稿日期:2004-12-21

作者简介:黄景文(1973-),男,广西来宾人,高级程序员,工程师,博士研究生,主要从事网络信息处理、决策支持等研究。

*国家计委西部大学网络工程项目(计投资20012437)资助。

流的第一行定义)之内,每段由信息头、空行、信息实体构成。由图 1 可知,上传的文件内容开始于 2 个回车换行符号之后,终止于边界分割符之前。

```
……
——————————7d421521b0292
Content-Disposition:form-data;name="filePub";filename="
D:\tt.txt"
Content-Type:text/plain

This is a TEST……
OK!
——————————7d421521b0292
……
```

图 1 通过浏览器提交上传的数据流

2 文件上传的后端处理

由上分析可知,要实现把文件上传到数据库,关键是从多域数据流中把文件内容析取出来,并写入支持 BLOB 的数据库中。其算法步骤一般为:(1)获取客户端表单提交的多域数据流;(2)提取边界分割符;(3)析取上传文件全文的内容(以二进制字符串表示);(4)链接数据库并打开数据表;(5)数据写入/修改;(6)关闭数据库。

2.1 数据库设计

选择 Microsoft SQL Server 2000 为数据库管理系统,该系统以 image 字段类型支持 BLOB。实际上,该类型是一种用来存储二进制数据的字段类型,用以存储图形图像、视频、音频、Word 文档、PDF 文件等非结构化二进制数据。为了实现文件上传到数据库,必须在相应的数据表中设计用于存放文件二进制数据的 image 类型字段,而且,为了便于用户在浏览器端的数据显示或下载,还需预留一个字段存放上传文件的 MIME 类型。例如表 1 和表 2 的相关字段分别用于存放教师基本信息和论文论著数据。

表 1 基本信息

字段名	类型	长度	说明
id	int	4	教师 id
Xm	char	20	姓名
Zp	image	16	照片
M_Type	char	40	照片类型

表 2 论文论著数据

字段名	类型	长度	说明
id	int	4	文件 id
id_A	int	4	教师 id
Content	image	16	文件全文
M_Type	char	40	文件类型

2.2 文件内容的析取

从图 1 可知,上传的文件内容开始于 2 个回车换行符号之后,终止于边界分割符之前。其实,根据 RFC1867 规范[2],多域数据流中每个数据段的实际内容都是开始于 2 个回车换行符号之后,终止于边界分割符之前。因此,用字符串定位并截取子串的方法可析取出表单各域上传到服务器的二进制编码。只要确知文件域在表单内各域中的相对位置,不难从截取到的相应子串得到文件全文。算法 1 为从多域数据流析取文件内容的算法,它假定表单中文件域前有 k-1 个表单域,则析取的文件内容保存在 data 数组的 k-1 单元,即 data (k-1)。

算法 1:GetFileData (k),传入参数 k 表示在客户端的表单中文件域前有 $k-1$ 个其他类型的数据域,结果由全局数组 data 存储各域数据。

步骤 1:算法初始化。从 Request 对象获取上传的多域数据流 msData;读取 msData 的第一行作为各域数据的边界分割符标志 diviver;计数器 i 置 0。

步骤 2: 定位到 msData 中的 2 个连续的回车符之后,即域数据的开始 dataStart。

步骤 3:在 msData 中截取从 dataStart 开始到下一个 diviver 之间的子串,赋给 data(i)。

步骤 4:msData←msData− data(i);$i \leftarrow i+1$。

步骤 5:如果 i≠k 则转步骤 2;否则结束。

算法 1 中用 VBScript 描述的 ASP 脚本,首先从请求对象中获取多域数据流的长度和二进制串,并提取边界分割符标志,之后依次提取各域,直到析取出文件全文 data ($k-1$)。

2.3 数据写入

采用 ADO 技术[3]可以方便地与后端数据库系统建立连接并实现对数据库对象的读写访问。

Recordset 记录集对象通过 2 个方法支持 BLOB 字段的操作,即 Recordset.Field.GetChunk 和 Recordset.Field.AppendChunk,其中前者用于数据读取(下载),后者用于数据写入(上传)。算法 2 为析取出的文件内容写入数据库的算法,算法接受某教师的论文论著全文上传,算法通过检测析取出的文件内容长度判别前端是否提交了论文论著文件,从而决定写入数据库与否,其中传入参数的意义分别为 id_A 表示论文作者的 id,m_Type 表示上传文件的 MIME 类型,binData 为从多域数据流析取出的文件内容的二进制字符串。

算法 2 WriteToDataPubs(id_A,mType,binData)。

步骤 1:算法初始化。创建 Recordset 对象的实例 rs,并打开待写入的数据表 Pubs。

步骤 2:调用 rs 的 AddNew 方法申请写入空间。

步骤 3:文件内容写入。如果 binData 长度大于 1,则调用 Appendchunk 方法将二进制数据写入数据表,执行如下操作:

rs("Content")←NULL

rs("Content"). Appendchunk binData

rs("M _ Type")←mType。

步骤 4:其他数据域的写入。如 rs("id _ A")←id _ A。

步骤 5:更新和关闭 Pubs 表,释放 rs。

3 应用分析

采用本文提供的方法实现了"广西大学教师信息网"整个网站资料的上传和管理。在实际应用中,具有如下优点:(1)实现了在浏览器界面上教师的结构化信息和非结构化信息的同步上传。其易用性使每个教师及时更新信息成为可能,使网站的数据来源即时、动态。(2)实现了长文本信息和带格式文本信息在数据库的存储,解决由于数据库系统的限制致使长文本被截尾或格式丢失的问题。(3)降低了系统管理员维护的难度。用数据库系统管理教师的结构化信息和非结构化信息,数据备份与还原简单。在我们的系统中,借助 Microsoft SQL Server 2000 的备份自动化工具,系统管理员几乎没有介入教师资料库的维护工作。(4)系统扩展、移植方便。详见表 3。

表 3 两种上传方法的性能对比

	基于 HTTP 协议和数据库的文件上传	传统的 FTP 方法
易用性	直接使用浏览器界面上传,上传的信息可编程。	需要客户端程序,上传后信息的可编程性差。
安全性	不需为用户开放后端文件系统的写权限,由数据库管理系统保证数据安全。	需为用户开放后端文件系统的写权限,存在安全隐患
同步性	结构化信息和非结构化的文件可在一张表单上同步上传,且统一存放在数据库。	结构化信息和非结构化的文件分开上传,前者存于数据库,后者写入文件系统。
维 护	数据备份与还原可借助于 DBMS 的自动化工具,系统维护简单。	系统维护须由手工完成。
扩展性	在关系数据库上轻易扩展	系统扩展须根据文件系统重新设计存储结构
移植性	只需迁移数据库	需保证文件系统的兼容
其 他	由关系运算保证数据的一致性和无冗余,数据存放集中。	文件按名存取,存储分散,易被重名覆盖和恶意修改,信息冗余和不一致、不可靠。

4 结束语

本文针对实际应用提出了一种基于数据库的文件上传方法,并在分析多域数据流的基础上给出了其实现算法。该方法应用于广西大学教师信息网站点的开发与运行,该站点由 Microsoft SQL Server 2000 支撑,投入运行半年以来,接受全校教师上传照片 1666 张,论文论著 15253 篇,2220 门课程的简介及教学大纲,这些数据将随时间递增。实际运用表明,该方法具有很好的效果。

致谢

韦化教授对网站开发及本专题研究提出许多宝贵意见,在此表示衷心感谢!

参考文献:

[1] 孙占东,姜加虎. 用 ASP 实现无组件的文件上传[J]. 计算机应用,2003,23(9):136-138.

[2] Nebel E,Masinter L. Form-based file upload in Html [EB/OL]. http://www.faqs.org/rfcs/rfc1867.html, 1995.

[3] 郑 章,陈 刚,张 勇,等. Visual C++ 6.0 数据库开发技术[M]. 北京:机械工业出版社,1999.

(责任编辑:黎贞崇)