Science & Technology Review 2007 Vol. 25 No. 16 (Sum No. 238)

# 基于 Web 的大文件上传技术研究

林天华[1,2],马素霞[1],赵 霞[2]

1. 华北电力大学计算机系,北京 102206
2. 河北经贸大学信息技术学院,石家庄 050061

[摘要] 基于 Web 的大文件上传是 Web 应用系统中经常遇到的问题。目前,基于 Web 的大文件上传组件比较缺乏,且不支持断点续传。在.NET 框架下,利用 JavaScript 和 ADODB.Stream 技术实现了文件的分块读取,利用 XMLHttp 技术实现文件块的异步发送,然后在服务器端接收文件块并记录接收日志,在此基础上实现了文件的断点续传,且传输速度较快,对文件的大小没有限制。

[关键词] XMLHttp; ADODB.Stream; 异步发送; 大文件上传; 断点续传

[中图分类号] TP393.093 [文献标识码] A [文章编号] 1000-7857(2007)16-0030-04

# Web Based Big File Uploading

LIN Tianhua[1,2], MA Suxia[1], ZHAO Xia[2]

1. *Department of Computer Science and Technology, North China Electric Power University, Beijing 102206, China*
2. *College of Information Technology, Hebei University of Economics and Business, Shijiazhuang 050061, China*

**Abstract**: Web based big file uploading is a common problem in web applications. At present, there are few web-based big file uploading components available, and none of them supports resumption from interrupted transfers. This paper implements a new big file uploading method, which supports resumption from interrupted transfers. File blocks are read using JavaScript and ADODB.Stream, and are transferred asynchronously using XMLHttp, and then log is placed on the server when received. The transfer speed is improved greatly, and there is no limitation for file size using this method.

**Key Words:** XMLHttp; ADODB.Stream; asynchronous transfer; big file uploading; resumption from interrupted transfers

**CLC Number:** TP393.093 **Document Code:** A **Article ID:** 1000-7857(2007)16-0030-04

## 0 引言

近年来,Web 应用系统越来越流行。一方面由于 B/S 结构安装部署简单,系统可维护性好,特别适合于多用户分布式模式;另一方面随着.NET,Java 等网络技术的发展,B/S 技术越来越成熟,优势更加明显。在 Web 应用系统中经常需要向用户提供基于 Web 的文件上传功能,如图片、视频、远程网络教育中的学生作业、老师课件笔记等。因此 .NET 平台和 J2EE 平台都提供了对 Web 文件上传的支持。

目前,国内外文件上传的 Web 组件已经很丰富,如国内的有纵横 HTTP 文件上传组件、化境无组件上传类、梁无惧无组件上传类等,国外的有 SlickUpload,AspUpload,SA FileUp 等[1-3]。但绝大多数组件只适合于上传小文件(30 MB 以下)。当文件较大时,大部分组件上传速度慢、不能实时显示上传进度;应用服务器对上传文件大小有硬性限制,如 IIS 限制上传文件最大不能超过 200 MB;最重要的是当前所有组件都不支持断点续传,而由于上传大文件持续的时间比较长,期间发生网络中断或计算机重启等意外的可能性较高,因此支持断点续传是上传大文件所必须的。

实现大文件上传可以采用 C/S 方式,如 ftp 形式,其优点是功能强大、上传速度快、能实现断点续传,但 C/S

收稿日期:2007-07-15

作者简介:林天华,北京市昌平区北农路 2 号华北电力大学计算机系,研究方向为软件工程、软件测试;
E-mail:lintianhua@heuet.edu.cn
马素霞(通讯作者),北京市昌平区北农路 2 号华北电力大学计算机系,教授,研究方向为软件工程;
E-mail:masuxia@jupiterst.com

方式部署麻烦、难以维护,不能实现与 Web 应用的无缝结合;也可以采用在网页上添加 ActiveX 控件方式(或 Java Applet),这种方式由于受到浏览器安全及权限的限制,能实现的功能不超过纯 Web 方式,且实现方式比纯 Web 方式复杂。

针对以上问题,本文在.Net 框架下,利用 ADODB.Stream 和 XMLHttp 技术研究实现了一种新的纯 Web 文件上传方法。经实验对比,本方法能上传任意大小的文件、支持断点续传、上传速度快、能实时显示上传进度,能很好地满足当前大文件上传的需要。

## 1 目前 Web 大文件上传的主要技术

在支持大文件上传的组件中,国外的 SlickUpload 性能较为优越,上传速度快,在局域网内平均上传速度能达到 3.5 Mbps (服务器和客户端都使用奔腾 2.4 GB,480 MB 内存),能显示进度条,上传文件大小不受限制。

SlickUpload 组件在浏览器端把上传表单 (包括上传文件)封装成一个 XML 文件,然后使用 XMLHttp 技术异步发送。为了突破应用服务器对上传文件大小的限制,SlickUpload 组件在服务器端使用 IHttpModule 技术在 ASP.NET 进程处理 request 请求之前截获 request 对象,然后使用隐含的 HttpWorkerRequest 对象通过分块读取和写入的方式来接收数据[4-5]。

```
provider = (IServiceProvider) HttpContext.Current;
wr = (HttpWorkerRequest)
provider.GetService(typeof(HttpWorkerRequest));
if (! wr.IsEntireEntityBodyIsPreloaded())
 {
      int n = 1024;
      byte[] bs2 = new byte[n];
      while (wr.ReadEntityBody(bs2,n)>0)
     { …}
 }
```

这种方式能针对大文件的上传特点对服务器接收过程进行优化处理,提高上传速度;在服务器端分块接收,能够实现大文件及超大文件的上传。然而,由于上传速度主要取决于网络速度,这种方式虽然在局域网内速度有很大提高,但在城域网和广域网中速度优势并不明显。最重要的是,该技术在浏览器端一次提交全部数据,因此不能实现文件的断点续传。

## 2 基于 XMLHttp 及 ADODB.Stream 技术的大文件上传方法

### 2.1 XMLHttp 及 ADODB.Stream 技术简介

XMLHttp 是传送 XML 格式数据的超文本传输协议。XMLHttp 的数据传输形式非常灵活:它上传的指令和下达的结果可以是 XML 格式数据,也可以是字符串、流,或者是一个无符号整数数组,还可以是 URL 的参数等。通过 XMLHttp 协议可以方便地在异构平台之间进行 XML 数据交换。XMLHttp 最大的用处是可以更新网页的部分内容而不需要刷新整个页面,因此它在浏览器端的应用非常普遍[6-7]。

目前,绝大多数浏览器都增加了对 XMLHttp 的支持,IE 中使用 ActiveXObject 方式创建 XMLHttp 对象,其他浏览器,如 Firefox,Opera 等通过 window.XMLHttpRequest 来创建 XMLHttp 对象。在浏览器端调用 XMLHttp 的过程很简单,一般有以下 5 个步骤:

1) 创建 XMLHttp 对象;

2) 打开与服务端的连接,同时定义指令发送方式,服务网页(URL)和请求权限等;

3) 发送指令;

4) 等待并接收服务端返回的处理结果;

5) 释放 XMLHttp 对象。

ADO(ActiveXDataObjects)是微软开发的数据访问组件,是一种提供访问各种数据库类型的链接机制。它很容易使用,并可扩展地将数据库访问添加到 Web 页面。其 Stream 对象是 ADO 2.5 及其以上版本所提供的新增对象,可用来表示文本流和数据流。它允许开发者对字节型的二进制流进行读写,因此可被用于实现客户端的读取文件功能[6],而且在微软各个版本的操作系统中都自带有 ADO 组件。

### 2.2 大文件上传的实现方案

实现大文件的上传可以分为 3 个步骤:

1) 使用 ADODB.Stream 装载文件并获取文件块;

2) 利用 XMLHttp 把文件块以 XML 形式异步发送出去;

3) 服务器端把接收到的 XML 文件解析成文件块并记录日志以备文件的续传。

发送文件的过程如图 1 所示。

若文件在上传过程中被意外中断,系统会在再次打开网页时,通过分析日志,提示当前用户文件未上传完毕。若用户决定续传文件,上传页面将根据日志中的

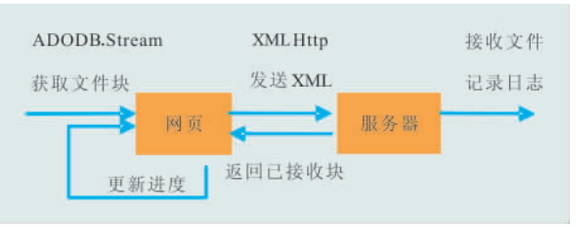


图 1 发送文件

Fig. 1 Upload file

文件名读取客户端文件，并根据已发送块和分块大小定位文件位置，使续传时只发送未上传的部分。断点续传的过程如图 2 所示。

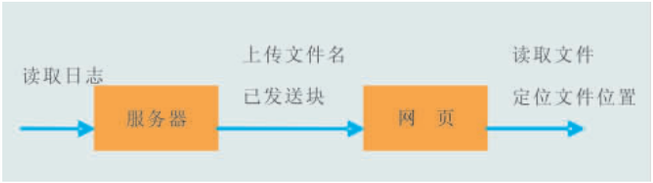


图 2 续传文件
Fig. 2 Resume broken uploads

### 2.2.1 上传页面的设计

用 VS2005 创建一个上传网页 Upload.aspx，在网页中添加 1 个 input file 控件 File1 用于选择文件；添加 2 个 input button 按钮 upload 和 continue，分别用于上传和续传；添加 2 个 input hidden 控件 fileName 和 startID，分别用于记录从服务器读取到的续传文件名和续传开始块。上传时，先通过 File1 控件选取要上传的文件，然后点击 upload 按钮触发 JavaScript 事件 BeginSendFile()。在事件中首先分析 File1 的属性，提取出被选择文件的完整路径名 vFullPath，然后开始装载文件和获取文件块。续传时，点击 continue 按钮，触发事件 Continue-SendFile()，根据 fileName 自动装载文件后，要先利用 startID 定位文件位置再开始获取文件块。

### 2.2.2 获取文件块

由于文件较大，全部一次上传的速度非常缓慢。为了实现文件的断点续传，必须对文件进行分块，分块的大小应根据计算机的速度和网络连接状况设置。分块太小会使总发送次数多，增加总数据通信量和服务器的处理负担；分块太大则会造成传完一块的时间较长，页面反应慢，续传效果不好。一般在 100 KB 到 2 MB 之间较为合适，如 BlockSize=1024×500，设为 500 KB。

获取文件块的第一步是利用 ADODB.Stream 根据文件名装载文件；第二步是定位文件流的当前位置，如果是上传新文件则为 0，如果是续传文件则根据从服务器日志读取的已发送块 startID 来定位（BlockSize* startID）；最后是从文件流中读取文件块。

```
// 创建读取文件的流
ado_stream = new
ActiveXObject("ADODB.Stream");
ado_stream.Type = 1;
ado_stream.Open();
// 读取文件
ado_stream.LoadFromFile(vFullPath);
//设置文件流位置
ado_stream.position = 0;
或 ado_stream.position = BlockSize * startID;
…
//从 ado_stream 中读取文件块
node.NodeTypedValue=ado_stream.Read(Size)
```

### 2.2.3 XMLHttp 异步发送文件

上传程序获取到文件块后，将文件块封装到 XML 文件中。首先，创建 XML 文件对象及节点并设置 XML 文件的发送格式，然后，将文件块赋给节点。除了文件块外，每次还应该发送文件名、本次发送的文件块的块号，是否为第 1 块或最后 1 块，以供服务器端根据情况处理。

```
// 创建要发送的 XML 文件
var dom=new
ActiveXObject("msXml2.DOMDocument");
var root=dom.createElement("root");
// 文件名属性
var fileName=dom.createAttribute("FileName");
// 分段开始标记
var bol_Start=dom.createAttribute("PartStart");
// 分段序号
var partID =dom.createAttribute("PartID");
// 分段结束标记
var bol_End =dom.createAttribute("PartEnd");
…
//把文件块赋予节点
node.NodeTypedValue=ado_stream.Read(BlockSize)
```

创建好 XML 文件对象后再利用 XMLHttp 发送 XML 文件。先创建一个 XMLHttp 对象；然后用其 open 方法创建一个新的 Http 请求。open 方法的第 1 个参数是请求方式，POST 表示发送，GET 表示获取；第 2 个参数是请求的 URL，这里指接收文件块的页面名；第 3 个参数是同步方式或异步方式，false 表示异步发送。发送完一个文件块后需要做一些处理，如更新进度和发送下一个文件块等，因此要设置 XmlHttp.onreadystate-chang =CallBack。CallBack 是一个 JavaScript 函数，可将发送完文件块后需要作的处理都放到该函数内。所有这些都设置好后，就可以把 XML 文件异步发送出去了。

```
//创建 XMLHttp 对象
XMLHttp=new ActiveXObject("Microsoft.XMLHttp");
//使用异步方式创建新请求
XMLHttp.open("POST","Recieve.aspx", false);
//XMLHttp 状态变化时调用 CallBack 函数
XMLHttp.onreadystatechange=CallBack;
//发送 XML 文件
```

```
XMLHttp.send(dom)
```

### 2.2.4 服务器端接收文件块、记录日志

服务器端接收文件时，首先创建一个 XMLDoc，然后用 XMLDoc 把输入流转化成 XML 文件，再从 XML 文件中解析出文件名、文件块号，最后写入文件块。

```
XMLDoc = new XMLDocument();
//从 Request.InputStream 装载 XML 文件
XMLDoc.Load(Request.InputStream);
…
//获取文件名
Name = node.Attributes["FileName"].Value.ToString();
//获取文件块号
PartID = node.Attributes["PartID"].Value.ToString();
//获取文件块的字符形式
string block = node.InnerText.ToString();
//从 Base64 编码转为二进制 byte 数组
ArBlock = System.Convert.FromBase64String(block)
```

如果是文件的第一块，则要创建新文件，如果是最后一块则要作结束处理，如删除对应的日志等。

```
if (bol_PartStart) {…}
if (bol_PartEnd) {…}
```

日志的作用主要是保存续传时所需的信息，因此所需要的属性是本次上传文件在客户端的完整文件名、当前上传的文件块号，如果允许客户端设置文件块的大小则还要记录本次上传使用的块大小。续传文件时，再从日志中把这些信息读到对应的控件上。

```
log = "Name=" + Name + ", PartID=" + PartID ;
bLog = System.Text.Encoding.UTF8.GetBytes(log);
fs = newFileStream(fn, FileMode.CreateNew);
fs.Write(bLog, 0, bLog.Length);
fs.Flush();
fs.Close();
```

## 3 结论

利用 ADODB.Stream 来装载文件并获取文件块，再通过 XMLHttp 把封装好的文件块发送给服务器的方法，能很好地解决大文件上传的问题。其上传速度快，且不论文件大小，在局域网内平均上传速度都能达到 3 Mbps（服务器和客户端都使用奔腾 2.4 GB、480 MB 内存）。由于使用了分块方式上传，能上传任意大小的文件，能实现文件的断点续传，突破了目前 Web 文件上传的各种限制。

## 参考文献(References)

[1] 徐欣欣. 基于 Web 的无组件多文件上传方案的研究与实现[J]. 计算机工程, 2005, 31(12): 232-233.
XU Xinxin. Research and implementation of multiple files upload with no third-party components based on Web[J]. Computer Engineering, 2005, 31(12): 232-233.

[2] 颜 波. 基于 ASP.NET 和 Oracle 数据库的图片上传和查看[J]. 计算机工程, 2005, 31(24): 207-209.
YAN Bo. Uploading and browsing images based on ASP.NET and Oracle database [J]. Computer Engineering, 2005, 31(24): 207-209.

[3] 佚名. 六款 Web 上传组件性能测试与比较[EB/OL]. [2006-08-26]. http://www.vxii.com/blog/article.asp?id=487.
ANONYMOUS. Performance test and compare of six file upload components [EB/OL]. [2006-08-26]. http://www.vxii.com/blog/article.asp?id=487.

[4] ANONYMOUS. SlickUpload overview [EB/OL]. [2007-03]. http://krystalware.com/Products/SlickUpload/.

[5] 崔国华. 磊客中国视频社区解决方案 [J]. 程序员, 2006(12): 68-69.
CUI Guohua. Solution of China Luike video community [J]. Developer, 2006(12): 68-69.

[6] 严海兵. 基于 XML 无组件文件上传的实现 [J]. 计算机工程, 2003, 29(4): 196-197.
YAN Haibing. Realization of XML-based and non-module file upload [J]. Computer Engineering, 2003, 29(4): 196-197.

[7] ANONYMOUS. AJAX file upload [EB/OL]. [2006-02]. http://www.codeproject.com/useritems/AJAXUpload.asp.

(责任编辑 代 丽)

## 更 正

由于作者和编辑工作失误，本刊 2007 年第 14 期第 78 页刊登的周建阳、朱诚的“美国《科学引文索引》和《工程索引》2006 年收录中国期刊统计”一文的表 1 中有两处错误：(1)第 84 项《中国科学(G 辑 物理、力学、天文、英文版)》的论文收录篇数应为 79 篇；(2)第 87 项《中国天文学与天体物理学报(英文版)》的“库别”应为 SCI。特此更正，感谢夏建白院士指出上述错误，并对所涉及的两个刊物编辑部及广大读者表示歉意。

(本刊编辑部)