

如何实现支持大文件的高性能HTTP文件上传服务器

blog.csdn.net

2017-08-12 21:48

HTTP文件上传是做Web开发时的常见功能，例如上传图片、上传影片等。实现HTTP文件上传也比较简单，用任何Web端的脚本都可以轻松实现，例如PHP、JSP都有现成的函数或者类来调用。但笔者最近在做项目时遇到了一个大问题，项目需要上传视频文件，这些视频文件的尺寸一般大于2GB，用PHP开发时，将服务器端的上传尺寸设置得足够大，但用Chrome、Firefox等浏览器上传时，经常出现响应超时而上传失败，怀疑是PHP的问题，后改用JSP来实现文件上传，现象也一样，经过分析后发现，原来PHP、JAVA的上传是先由服务器缓存为临时文件，或者服务器将上传数据缓存到内存中后，再由脚本调用相关的上传文件处理函数来移动临时文件来保存文件数据；由于PHP、JAVA等处理文件上传需要分两步，对于大文件与超大文件来说，再次移动文件也是比较耗时间与系统资源的，由于浏览器将文件提交到服务器上后就会等待服务器端的响应，服务器端移动文件耗时太长，导致浏览器等待超时而报错。看来采用这种方式无法解决这种问题，于是决定重新自己实现一个独立HTTP文件长传服务器来供Web应用系统调用。

一、HTTP文件上传的技术原理

HTTP文件上传是通过 multipart/form-data 协议实现的，multipart/form-data实际上是一种数据的编码分割方式，例如在浏览器端编写一个文件上传的页面，向服务器发送POST请求后，服务器端将会收到如下数据：

```
POST /douploadfile HTTP/1.1 Host: www.melody.com:8068 Connection: keep-alive Content-Length: 510 Cache-
Control: max-age=0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*.*;q=0.8
Origin: http://www.melody.com:8068 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/39.0.2171.65 Safari/537.36 Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary4Ze9aQ5NKOuihnkZ Referer: http://www.melody.com:8068/douploadfile Accept-
Encoding: gzip, deflate Accept-Language: zh-CN,zh;q=0.8 Cookie: aa_videos_per_page=30;
guest_name_43b2525ed9f77823e78012a4f24405a5=admin -----WebKitFormBoundary4Ze9aQ5NKOuihnkZ
Content-Disposition: form-data; name="file1"; filename="file1.c" Content-Type: text/plain
1111111111111111111111111111111111111111111111111 -----WebKitFormBoundary4Ze9aQ5NKOuihnkZ Content-
Disposition: form-data; name="file2"; filename="file2.c" Content-Type: text/plain
222222222222222222222222222222222222222222222222222 -----WebKitFormBoundary4Ze9aQ5NKOuihnkZ
Content-Disposition: form-data; name="text1" ABCDABCD -----WebKitFormBoundary4Ze9aQ5NKOuihnkZ--
```

multipart/form-data 协议将文件数据与Form的表单数据一起传输，通过分割符来区分表单域与文件数据，这个分割符是在HTTP请求头里指定，例如本处的分割符为

oundary=----WebKitFormBoundary4Ze9aQ5NkoUihnkZ，每个分割符前有个分隔符头 "--"，最后一个分割符也后也必须添加 "--"。只要知道HTTP上传文件的数据格式，就好实现文件上传服务器了，

只要在接收完毕HTTP请求头后，将后续的数据通过分割区分即可，如果是文件数据，分隔符后的附加数据会指定 filename 名。

二、技术实现

出于性能考虑，笔者实现的HTTP文件上传服务器采用C语言实现，文件上传服务器是作为独立进程来运行的。采用独立文件上传服务器上传文件的过程是这样的：

1) 浏览器将文件上传请求提交到 文件上传服务器。

2) 文件上传服务器在接收文件完毕后将文件的信息例如长度，物理位置与访问URL返回给浏览器，通过JSON方式即可。

3) 浏览器接收到上传的反馈信息后暂存，然后再与其它信息一起提交到Web服务器。

由于文件上传服务器与Web服务器不是同一进程，监控的TCP端口不同，在向上传服务器发起POST请求时，浏览器将会认为是不同的域，会出现跨域问题，浏览器在遇到跨域访问时，一般会先发送OPTION请求，上传服务器需要对跨域做出正确的响应，在响应头里加入 Access-Control-Allow-Origin: * 即可。

文件上传服务器采用JSON来反馈信息。

文件上传服务器采用线程SELECT模型，支持1000个并发连接，支持大于2GB的文件上传，采用即时写入技术，不需要再次移动文件。

采用无刷新界面方式来上传文件，即发送上传请求后，主界面无需刷新。

文件上传服务器支持HTML5方式长传，并能够显示上传进度，由于这是通过浏览器本身来实现的，比较容易。

文件上传服务器支持传统浏览器（例如IE8或者以下版本）上传数据，也能够显示进度，通过AJAX定时向服务器发送进度查询请求来实现。浏览器在发起上传请求时，制定一个上传ID，然后定时通过这个ID来获取上传进度。

由于上传服务器的代码足够多，因此不——阐述，请查看附件的文件上传服务器样本。

三、浏览器端调用

.1 HTML5文件上传调用

HTML5上传是采用 HTML5的FormData 与XMLHttpRequest来实现的，具体如何操作请查阅相关资料。步骤如下：

1) 创建FormData并将原始form表单的DOM元素作为一个参数传递给FormData；

2) 创建XMLHttpRequest对象并设置好参数

3) 调用XMLHttpRequest对象的open方法并制定正确的URL，这个URL应该是上传服务器的接受地址。

4) 调用XMLHttpRequest的send方法。示范代码如下

```
var myForm = document.getElementById("form1")
```

```
var fd = new FormData(myForm)
```

```
var xhr = new XMLHttpRequest()
```

```
xhr.upload.addEventListener("progress", uploadProgress, false)
```

```
xhr.overrideMimeType("application/octet-stream")
```

```
xhr.addEventListener("load", uploadComplete, false)
```

```
xhr.addEventListener("error", uploadFailed, false)
```

```
xhr.addEventListener("abort", uploadCanceled, false)
```

```
xhr.open("POST", upload_url,true)
```

```
xhr.send(fd)
```

2 IFRAME文件上传调用

由于微软的IE8及以下浏览器不支持HTML5，我们旧的采用传统的方式来上传文件。为了实现界面的优雅设计，即无刷新方式上传文件，在进行传统方式上传文件时，将 form 的 target 属性指向一个 iframe，浏览器将在上传完毕后将返回网页输出到这个iframe，通过将这个iframe隐藏，可以避免破坏界面的美观。

iframe方式上传文件的步骤如下：

- 1) 构建一个隐藏的iframe。
- 2) 设置form的target为网页里的一个iframe的名字一边指向这个iframe。
- 3) 用普通的submit提交表单。
- 4) 从iframe里获取返回信息 iframe方式也适用于现代浏览器，包括移动设备浏览器。

采用iframe方式提交数据后，在数据提交结束并获得反馈后，会触发iframe的load事件，我们通过设置iframe的load事件来获得反馈。由于访问 iframe与应用服务器仍旧是不同的域，需要绕过这个跨域访问 限制，iframe的name属性可以作为数据交互的纽带。HYFileServer服务器通过产生相关代码来修改iframe 的name属性，父框架通过访问iframe的name属性来获得反馈信息。

如果需要测试，可以通过 [github](https://github.com/wenshui2008/HYUploadServer)来下载 这个文件上传服务器

<https://github.com/wenshui2008/HYUploadServer>