

The Implementation of Large Video File Upload System Based on the HTML5 API and Ajax

Yungeng Xu^{1, a}, Sanxing Cao^{2, b}

¹Department of Science and Engineering, Communication University of China, Beijing

²New Media Institute, Communication University of China, Beijing

^axuyungeng1016@163.com, ^bcaosx_cuc@126.com

Keywords: HTML5; File upload; Shell; XMLHttpRequest

Abstract. The function of upload file is an important web page designs. Traditional file upload function for file upload function by HTML form and \$_FILES system function, has the advantages of convenience, can cope with a relatively small file upload. The traditional HTML way has been difficult to meet the upload of the oversized file. Service not only to open a link waiting for this file is uploaded, but also to allocate the same size of memory to save this file. In the case of large concurrency may cause a great pressure on the server. This article is the use of HTML5 File API to upload video files in the original form on the basis of form, use the function of slice inside the HTML5 file API for the split file upload, to achieve the result of no refresh upload files. Have a valid practical significance.

Introduction

With the development of web technology, watch videos on the website has become an important forms of entertainment in the people's daily life. Internet video give people a great deal of freedom to choose the way of entertainment, not limited to the live TV mode. People can watch the video on the webpage. Webmasters corresponding need to upload video files to work in the background.

The traditional video upload is through the HTML form and \$_FILES system function or use PHP + flash to upload file. The advantage is convenient, the disadvantage is that there are size limits, for big video file upload limit, the speed also to have certain restriction.

This article is in the form of traditional forms, through HTML5 File API and JavaScript API cut video file to upload, and then composite video through Shell Script in Linux systems. You can upload large video files, and has no effect refresh upload with upload progress bar. Has a good interface experience. It can provide a traditional video files uploaded effective means of settlement.

The video file is used as a unit of data in external memory set. And an image file storing the character itself, sound, video and other media file information, are the successive binary data via the input/output device was subjected to coding processing, which itself could not influence the basic form of stored content. So the video file is a binary file.

Cutting through read video files, binary stream, byte binary stream will completely according to the procedures for the preparation of their form is written to a file or device, and completely according to them from the file or device reads form read into the program. Encoding each part of the cut is not impaired, the final synthesis of video is no different compared to the original video.

System request

This system is based on HTML 5 and Ajax in design, need a browser support HTML 5 and the new version of Ajax technology. As the HTML 5 standard have been developed, now the major browsers generally support the new version of HTML 5 technology, such as “Firefox, Chrome, Safari, IE 10”, etc.

The system designed in this paper by using Linux Ubuntu as a server, the following reasons: video synthesis requires shell script support in Linux system. Subsequent to the video processing using FFMPEG to processing, need the support of Linux.

PHP Call. The system uses web technology development, the background video processing by calling PHP shell script, PHP as a server-side scripting language, invoke external commands in PHP. In this paper, we adopts the `exec ()` function to call shell script.

Upload progress. In order to detect the upload progress in real time, we use the characteristics of Ajax, by uploading a callback upload progress in real-time monitoring of the effect achieved the result of no refresh upload. Let it on the page can real-time display the status of the progress bar.

System implementation

This article is based on web video upload. Mainly to implement video cutting, upload video, video synthesis module. The specific process as shown in figure 1.

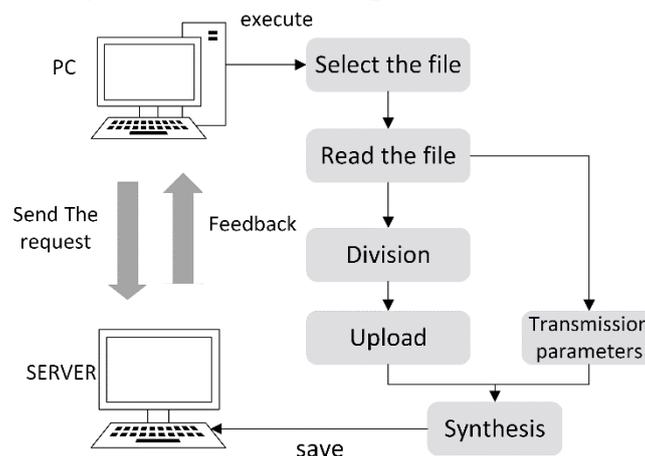


Fig. 1. The flow chart of system

Video cutting. The new features of HTML5 allows the browser to access the file system has limitation, the new browser allows JavaScript existing file input field in a form of file access. It can be used directly by the data in JavaScript file. JavaScript increase blob data types and interfaces for dealing with binary large objects. Blob can be read from a file or written to the file. Blob object represents a browser can read a set of raw binary stream. Blob object, property size represents the size of the stream. The slice File object () method accepts two parameters, the starting byte and bytes to read. The method returns a blob instance. Blob type attributes have a size and type properties, and also has a slice () method to extract part of the blob.

In HTML 5 File API allows us to read local files. File object usually by clicking the “type=file” input form, HTML 5 drag and drop API to get FileList array. By pasting in the input box, which contains one or more of the File object. The form contains a file upload fields will provide a FileList object. The slice () method to read the section for file segmentation.

The system is set to read a byte is 1048576 bytes, which is the size of 1M.

Video upload. The new version of XMLHttpRequest interface provides functionality to upload files to the server. By using FormData interface, the program may be packaged files and sends them to the server. Using FormData we can through JavaScript with some key value pairs to simulate a

series of form controls, we can also use XMLHttpRequest the “send()” method to asynchronously to submit the form.

FormData can pack data into standard http file upload server is uploaded, you can put the relevant parameters such as file names and file contents as a blob passed to FormData.append (), injected into the value of the field to FormData object. We can artificially add a key value to the current FormData object. Then use XMLHttpRequest interface to send as Ajax FormData object to the appropriate procedures for processing. The key code is as follows:

```
var eof = 0;
xhr = new XMLHttpRequest();
fd = new FormData();
file = document.getElementById('fileToUpload').files[0];
```

The “eof” Which is defined to identify the upload is complete, when the file upload is complete, make it to 1, for further processing.

Then the FormData object as a parameter to the XMLHttpRequest object the send () method, using XMLHttpRequest the open () method to open a request to pass the parameters need to be stored to the next program for processing:

```
fd.append ("partnum", start/range);
fd.append ("token", token);
fd.append ("eof", eof);
fd.append ("ID", id);
xhr.open ("POST", "upload.php");
xhr.send (fd);
```

The “partnum” is the number of copies of video segmentation in video, token as a temporary filename. The segmented video named according to “partnum” named “token-partnum”. The “ID” is the corresponding video id number which saved in the database used as the name of the video after synthesis.

Open the browser tool to view the POST request as shown in figure 2.

```
▼ Request Payload
-----WebKitFormBoundaryRMuX9sgM8sNG2Qe2
Content-Disposition: form-data; name="part"; filename="blob"
Content-Type: application/octet-stream

-----WebKitFormBoundaryRMuX9sgM8sNG2Qe2
Content-Disposition: form-data; name="partnum"

14
-----WebKitFormBoundaryRMuX9sgM8sNG2Qe2
Content-Disposition: form-data; name="token"

55150d24a8c36
-----WebKitFormBoundaryRMuX9sgM8sNG2Qe2
Content-Disposition: form-data; name="eof"

0
-----WebKitFormBoundaryRMuX9sgM8sNG2Qe2
Content-Disposition: form-data; name="ID"

238
```

Fig. 2. The network request in firebug

XMLHttpRequest also provides some callbacks, you can provide feedback to the user. Through monitoring the upload the progress of the event object. The “onprogress” event returns the number of bytes and the total size of the upload has been sent, you can display a progress bar design according to the callback.

Video compositing. Through XMLHttpRequest open () method to open request, transmit the video data to the synthetic procedures for processing, parameters are sent by the post method to obtain. Judgment to the value of “eof”, when its value is 1. Use the exec () method call shell script

for synthetic video command.

```
if ($eof == 1){ exec("/var/www/chinavec/upload/cat.sh '$partNum' '$token' '$ID' > /dev/null &"); }
```

In the shell script, use the Linux “cat” command to video synthesis. The “cat” command for the Linux of the command system, can be used to splice composite file:

```
for ((i=1; i<=$partNum; i++))do echo $token-$i; done | xargs -i cat {} > /var/www/upload/tmp/$token.mpg
```

Here we use the “for” loop function merge split video files into MPEG format in sequence.

Test and verify

Upload test. Through the video upload to server in the local, the progress bar can display real-time schedule. The video file also can upload to the server. As shown in figure 3.

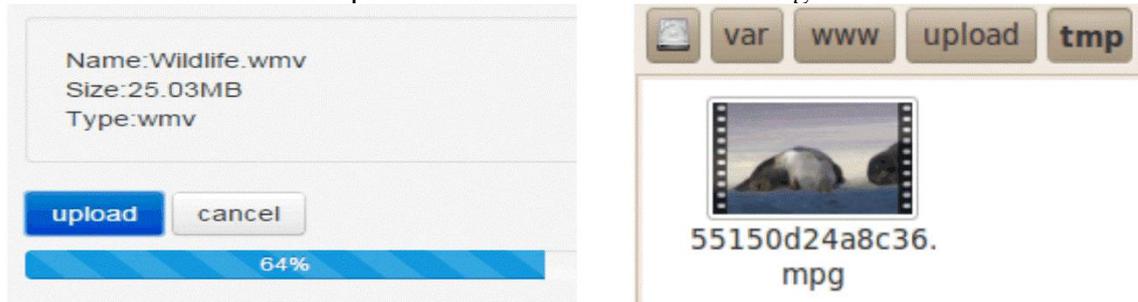


Fig. 3. The view of the webpage and upload results

Summary

In order to solve the traditional web page upload limitations, this paper is through the HTML 5 API and Ajax technology, use Linux Ubuntu as the server, at the same time, using PHP and MySQL to develop web page upload video system. Upload mainly implements video segmentation, video, video synthesis module. Test results show that it can realize the function of upload large files, and can display real-time progress bar. It provides an effective video upload method in the webpage.

References

- [1] Xiaojin, Chen. “Research on file upload based on HTML5,” in 2014 11th International Conference on Service Systems and Service Management (ICSSSM2014), pp. 1 – 3, doi: 10.1109/ICSSSM.2014.6874102
- [2] M. Casario, P. Elst, C. Brown, N. Wormser, C. Hanquez, HTML5 Solutions Essential Techniques for HTML5 Developers, 2011, pp. 195-192.
- [3] Z. Kessin, Programming HTML5 Application, 2012, pp. 66-72.
- [4] P. Lubbers, B. Albers, F. Salim, Pro HTML5 Programming , 2010, pp. 169-191.
- [5] A. Ranganathan, J. Sicking. File. (2013, Dec.) File API W3C. [Online]. Available: <http://www.w3.org/TR/FileAPI/>
- [6] M. Niedermayer , S. Sabatini, V. Giovara. (2012, May.) FFmpeg Documentation. [Online]. Available: <http://ffmpeg.org/ffmpeg-all.html#Video-Encoders>
- [7] R. Blum, C. Bresnahan, Linux Command Line and Shell Scripting Bible, 2008.

[8] Xu Xinxin, Ai Fei, “Wang Yanxuan. Research and Implementation of Multiple Files Upload with No Third-party Components Based on Web”, Computer Engineering, June; Vol. 31, No. 12, pp232-234, 2005.