# The broken-point continuingly-transferring scheme of large files based on HTML5

Limin Wang[1, a *], Zhenghe Liang[2,b] and Quanfeng Duan[3,c]

[1]School of Computer and Information,Hohai University,Nanjing 211100,China

[2]School of Computer and Information,Hohai University,Nanjing 211100,China

[3]School of Computer and Information,Hohai University,Nanjing 211100,China

[a]15751872027@163.com, [b]zhliang@hhu.edu.cn, [c]duanquanfeng@163.com

**Keywords:** large file uploading;HTML5;FILE API;broken-point continuingly-transferring;files merging;

**Abstract.** In Web applications, it is often needed to upload a file to the server. With current file upload methods, it is difficult to deal with large file uploading and user experience is also bad. Uploading big files often failed because of network interruption and the client had to reupload.With the development of HTML5 technology,a series of API about file operation emerged.This makes it possible to use javascript on the client side to slice local files and further achieve the function of file broken-point continuingly-transferring.On the basis,this paper solves timeout problem of merging files and correctness problem of the final file on the server side.

## Introduction

Uploading large files takes relatively longer time in the current network environment.Power loss and network interruption is inevitable during the progress.Only the partial is uploaded successfully and then the process is interrupted,causing the file upload failed.In the absence of broken-point continuingly-transferring functionality,the system has to re-upload the file from the beginning.This makes massive duplicate data transmitted in the network,wasting the bandwidth and increasing the time.

The current methods to achieve file broken-point continuingly-transferring are mainly based on FTP technology.But many universal firewalls on the internet do not allow FTP to upload files,only allowing www server.So developping file uploading system based on http protocol is necessary.There are many good downloading softwares based on http protocol now.But uploading softwares based on http protocol is a vacancy and have many defects,not satisfying social needs.

To solve these problems, we propose a scheme based on html5 to implement large files broken-point continuingly-transferring.The scheme is based on http protocol,solving the problem of large files broken-point continuingly-transferring and enhancing the uploading efficiency.In this solution,the web side slice uploading file using the slice method of html5 and transfer file fragmentation to the server side using Ajax.When the server side receives fragmentation data,append the received data to the corresponding temporary file.If the network interrupts during the uploading progress ,the web side makes the request of re-uploading to the server side.Then the server side computes the broken-point location and send it to the web side,specifying this location for the starting position of uploading.The web terminal will begin to send fragmentation data from this location.

## HTML5 and the related file operation API

HTML 5 is the next generation of HTML.World Wide Web Consortium which is the most authoritative and influential organization in the web technical realm completes to formulate its standard specification on October 29, 2014.Compared to the previous version,there are many simpler and more semantic tags.The new version enhances the function of the label,has the powerful processing capability in multimedia applications and supports offline storage, multithreading and Geolocation.The related FILE API used in this paper is also part of the specification.

 

HTML5 provides a series of interfaces about processing files,including Blob,File,FileReader and FileList.Blob interface represents raw binary data, and allows access to ranges of bytes within the Blob object.File interface includes readonly informational attributes about a file such as its name, its mediatype, and a URL to access its data.FileReader interface provides methods to read a File, and an event model to obtain the results of these reads.A FileList sequence, which represents an array of individually selected files from the underlying system. The user interface for selection can be invoked via <input type="file">, i.e. when the input element [HTML5] is in theFile Upload state.

1.FileList,an array of user-selected files.The client selects files using the input tag <input type=file" multiple/>.The file list has a readonly attribute length which means the number of file objects and a method to obtain specified file object.

```
var fileList = document.getElementById("file");
var file = fileList.files[0];//obtain specified file object;
```

2.File,extends Blob.In addtion,it has some additional readonly attributes such as file name,file mediatype and file URL.

```
var fileSize=file.size;//obtain file size in bytes
var fileName=file.name;//obtain file name
var fileType=file.type;//obtain file madiatype
```

3.Blob,represents raw data. It provides a method to slice data objects between ranges of bytes into further chunks of raw data and an attribute representing the size of the chunk of data.The slice method returns a new Blob object between the ranges of bytes specified. It is defined as follows:Blob slice(int start,int length).The start parameter is a value for the start point of a slice call.The length parameter is a value for the end point of a slice call as byte offsets from start.We can use the slice method to slice large files into many small chunks.Processing these chunks is equivalent to processing the large file.

```
var chunkSize=5*1024*1024;//set the size of each chunk is 5M
var numOfChunks=Math.ceil(fileSize/chunkSize); //compute the number of chunks
blob = file.slice(i * chunkSize, (i + 1) * chunkSize);
```

These chunks are sent to the server side using Ajax

## Implementation of large file broken-point continuingly-transferring

The so-called broken-point continuingly-transferring is that the client can upload files from the broken-point ,not from the beginning afer the progress of uploading large files is interrupted.

On the server side,there are two strategies to merge file fragmentations.One is to merge after all the chunks are already sent to the server.When the server side receives a file chunk,write it into a new temporary file.Each chunk,each temporary file. When receiving the request for merging files,merge all the temporary files according to the order into a file and delete those temporary files.In this method,if the file is large enough ,the server side will generate massive temporary files,resulting in too long combined time and even exceeding the time watiing for response.The other is merging while transferring.The server side establishes a temporary file after receiving the request of starting to upload files.Once the server receives a file fragmentation,append it to the end of the corresponding temporary file.When receiving the request that file has already been sent completely,upload file successfully.The approach solves the problem of waitting too long on the web side.But during the continuingly-transferring progress,there are extra data in the temporary files,forming junk files.This paper selects the second method.To prevent generating junk files,set a buffer on the server side.The buffer length is equal to the chunk size on the web side.Write a complete fragmentation into the corresponding file while the transferring progress is not interrupted,or abandon the already received partial data and do nothing.In addition,this paper uses file checksum to ensure the validity of final merger, thus improving user experience.

Set a buffer whose length is greater than or equal to the chunk size set on the web side.The web side upload files using AJAX.There are three kinds of data transfered to the server.The first is start type,indicating that the client begins to transmit data.In the transmission parameters,set file name and

file size.The server determines whether the file has been uploaded or not in the light of file name.If this file was uploaded once,the handler on the server side computes the broken-point position and returns it to the client,informing to send file from the position.Or creat a temporary file and return 0 to the web.The second is data type,explaning the transfered is fragmentation data.When the server receives the request marked with data,the handler reads file stream data and write it into the buffer.After completing reading,append the buffer data to the corresponding temporary file and return success response to the web.If the progress of reading fragment data is interrupted,abandon data in the buffer.It ensures that every deposit is a complete file fragmentation.After receiving the response,the web continues to transfer the next chunk until all the chunks have already been sent.The third is end type.After the web receives success response of the last fragment,send end request to the server,informing that file upload has already completed.After the server receives the request, the handler checks whether the size of the temporary file is equal to file size that is a property of file.If they are equal,rename temporary file and return success response.Or reestablish the temporary file and return reuploading response.If the upload fails,the client relogin and send request to the server.Fig.1 demonstrates the above progress of implementing broken-point continuingly-transferring.

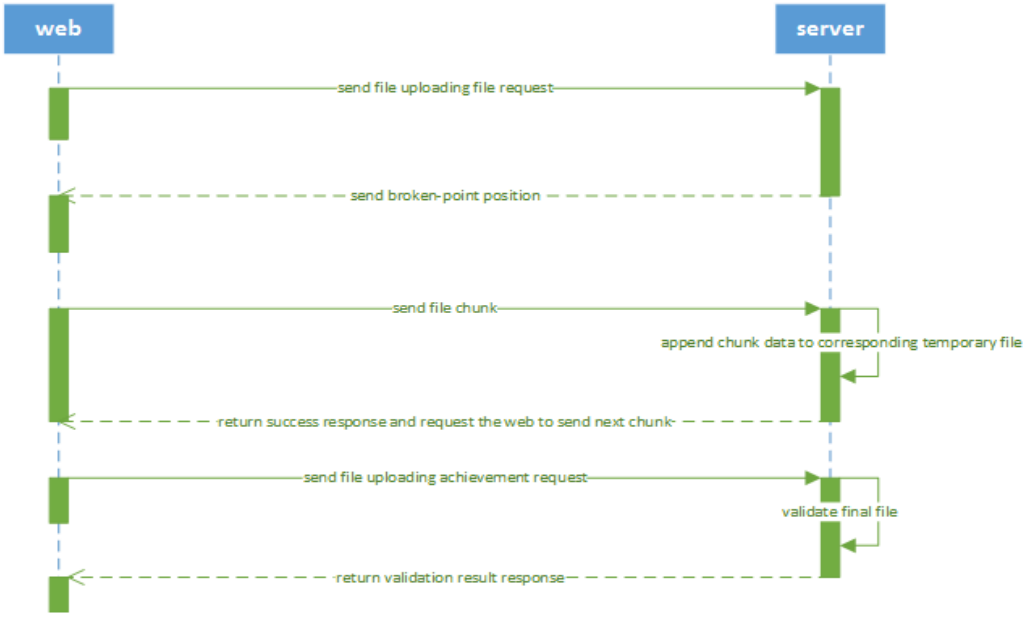


Fig.1. Implementing broken-point continuingly-transferring

1.the transmition of data in start type

The client selects file to upload and then send start data to the server.The data is ordinary data for cummunication with the server, not the chunk data about file content.The handler vertifies the file according to file name and computes the broken-point,then returning it to the web.The web begins to upload fragmentation at the broken-point.

The calculation for broken-point is relative to temporary file’s principle of management towards fragmentation.The temporary file is updated in denomination of fragment.So the length of temporary file is integral number of times as long as chunk size.Set the temporary file as file,the size of each chunk as chunkSize,then the broken-point is file.length()/chunkSize.

2.the transmition of data in data type

The web side starts to send chunk data from the broken-point which is returned from the server.At the same time,the client transmits some meta data,including file name and chunk size.The server searches the corresponding temporary file according to file name.It reads file stream data and write it into the buffer.After completing reading,append the buffer data to the above temporary file and return

success response to the web.If the progress of reading fragment data is interrupted,abandon data in the buffer.

Set a buffer whose length is greater than or equal to the chunk size of the web side.The buffer is used to load file stream data from request.In addition,set another buffer named buf.In this paper ,set the length of buf 16*1024 byte,that is 16kb.Read a buf from file stream every time and then write into buffer until the end of file input stream.If the web terminal is interrupted during reading process,abandon these data.Or append the data to the temporary file. Fig.2 demonstrates the transmition progress of data in data type.

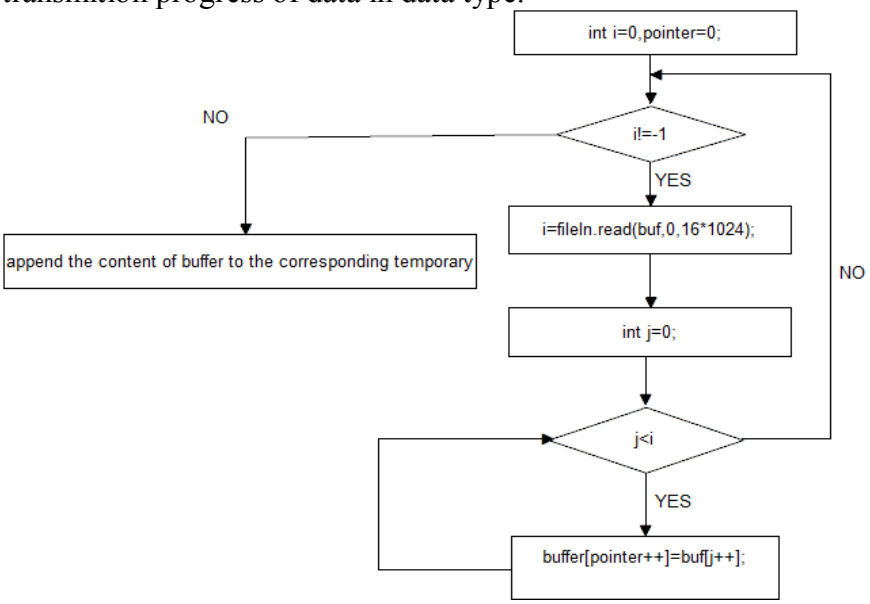


Fig.2.The transmition progress of data

3.the transmition of data in end type

The web side sends end data to the server,indicating upload is complete after all the fragmentations have already been sent out.Request parameters include full file name,file name,total number of chunks,file size.The handler on the server obtains file name,full file name and file size through request object.Judge whether the size of the corresponding temporary file is equal to value of parameter filesize.If they are equal,rename temporary file to fullname with file suffix.Now file upload succeeds.If not,delete the temporary file and return failure response to the web.

## Analysis

The continuingly-transferring scheme adopting real-time consolidation merge files while receiving file fragmentation.It can prevent too long waiting time for the client caused by consolidated combination and thus improves user experience.Set buffer on the server side and write fragmentation data into the buffer.If the reading progress succeeds,append data in the buffer to temporary file.Or abandon these data.It not only ensures that there are not repeated dirty data in the file,but also ensures the correctness of uploading file.

## Conclusions

The slicing method breaks through the restriction on upload size and provides a new idea to achieve continuingly-transferring.The web side can upload file of any size in theory.The continuingly-transferring method using File API provided by html5,solves the failure problem of uploading files because of too large file or network interruption.The tactics of real-time consolidation settles the trouble of waiting too long.It improves user experience greatly.

**Acknowledgements**

At the end of the paper,I would like to express my sincere thanks to many people.First of all,I am grateful to my supervisor,Mr Liang.He gives me constant encouragement and guidance during the progress of accomplishing this thesis.Without his instruction,the thesis could not have reached its current level.Secondly,I'd like to express my gratitude to my classmate.He gives me a lot of inspiration in our communication process.Last but not least,I am indebted to those leaders, teachers and working staff for their work to my thesis.

**References**

[1] RFC1867 - Form- based File Upload in HTML[S], 1999.

[2] Information on http://www.w3.org/TR/FileAPI

[3] Xiaojin C. Research on file upload based on HTML5[C]//Service Systems and Service Management (ICSSSM), 2014 11th International Conference on. IEEE, 2014: 1-3.

[4] CHEN Z, GUO J, LIU Z, et al. Design and Key Technology of the File-transfer System with the Function of Broken-point Continuingly-transferring [J][J]. Computer Engineering, 2002, 12: 005.

[5] Yuanwen Li,Mingzhi Cheng,Xiuhua Xu,et al.The applied research of broken-point continuingly transferring and multithreading in remote transmission version in Chinese[J]. Journal of Beijing Institute of Graphic Communication, 2012.

[6] Xin Meng,Research and implementation of efficient file upload method based on FTP in Chinese[D].South China University of Technology,2014