File Uploads - Does Size Really Matter?

Posted At : January 6, 2006 3:00 PM | Posted By : Mark Kruger
Related Categories: Coldfusion Optimization, Coldfusion Troubleshooting

When creating complicated web applications you sometimes run into situations where long running requests are *necessary*. Please don't email me with altruistic best practice jargon. I *know* they are not a good idea. My point is, that they can be unavoidable in some cases. Consider file upload for example. Suppose you need to allow users to upload more than 1 file in a single request (let's say 5) and suppose the files are typically .5 to 2 Megabytes. Potentially this means a maximum total aggregate of 6 megabytes.

That's certainly not the *worst* size problem I've heard of - in fact, my own size problem is much worse if my spam folder is to be believed... but I digress. This 6 megabytes will create a long running request. Even on a high bandwidth connection it *could* take a couple of minutes for various reasons - right? On a low bandwidth connection it is going to take *several* minutes. What are the implications?

User Impatience and the dreaded Double Submit

I'm an advocate of handling things on the server. When folks ask me "should I validate on the client or the server" my standard answer is that if you have to choose, do it on the server. Server side validation protects the data. Client side validation makes for a better user experience, but does very little to protect the data. Some combination of *both* is usually the right way to go. Can you handle a double submit on the server? Sure - added checks to the database, session, etc will help you ensure the user information is not submitted twice.

But the case of our file upload brings up a separate issue. In this case submitting again is not a trivial matter because it uses up *valuable server resources*. So the first thing to do when the necessity to create a long running request rears it's ugly head is to disable the submit button or use some other client side JavaScript to restrict the client from submitting again. In fact, this is typically very easy and should be done in *most* cases.

Providing "Upload Progress" Feedback

This is a much tougher issue. You see, in order to provide feedback *from the server* you will need to wait till the entire request is completed. The nature of HTTP is such that the server is informed of the request size in advance and it waits till all the bytes have come in before it runs it's routines. So useful feedback *during upload* is impossible using regular HTML or JavaScript.

There are other alternatives - all of which monitor the upload from the *client*. For example there is a flash upload component that provides a nice progress bar. There are a number of Java applets for this purpose as well. As you contemplate one of these nifty solutions keep in mind that both your support and user sophistication must increase in order to make use of it. You might be better off simply posting sufficient warnings to folks using the form - "hey - it could take several *minutes*, so keep your shorts on!!" (that's a message for non-paying customers of course).

Response Pages and White Space (new discovery)

Recently we discovered that if a request ran a long time, and if the action page produced a lot of white space, and if the firewall was configured a certain way, and if the earth was aligned with Uranus, there was a problem displaying the results of a file upload post request. The client would see the status bar and wait and wait... nothing happening. The request would actually *finish without error* on the server which could only mean that it was successful as far as the web server knew. The user, however would never see the results or action page content.

The Fix

It turns out this was only a factor if the result page was displayed as a direct result of the post. If the post implemented a redirection (CFLOCATION) after the event was handled, everything worked. This is because Cflocation clears the buffer and sends a new location back through the response header.

So my final advice is - use CFLOCATION on responses that are the result of a file upload. It will save you headaches, and it will also make it impossible for folks to click on the "refresh" button at the top and reposting the data.

Let's hear about your best file upload tips. I can take it :)