

Looking for Bottlenecks? Watch Out for Cfdirectory

Posted At : October 25, 2006 7:36 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Optimization, Coldfusion Troubleshooting

I have a client with a file intensive application. It allows users to upload images and manage galleries. It's very slick and uses the flash uploader to accomplish multiple file uploads. He was having performance problems with the uploader. The flash uploader is a nifty way to upload a wheel-barrow full of files in a single operation. You can even check for things like file size and type *in advance* instead of waiting for the whole file to arrive on the server.

What we began to notice is that some requests took longer than others, *a lot longer*. I, being the expert troubleshooter that I am, naturally thought it was file sizes. I assumed that requests for a 2 meg file upload naturally took longer than requests that handled files of 200k. When we looked closer, however, it turns out that was *not* the case. A much more sinister culprit was lurking.

The problem

It turns out the file handler does a number of things. It uses the "upload" action of cffile to store the file in a temporary location, then it checks the main file directory for the user to see if there is already a file with that name. If there is, it uses a short routine to create a new filename. I created a little "time" structure that I emailed myself on every successful request. Like so:

```
<cfset times = structNew() />
<cfset tm = getTickCount() />

... file upload code

<cfset times.fileupload = getTickCount() - tm />
<cfset tm = getTickCount() />

... cfdirectory code to get list of files

<cfset times.dirList = getTickCount() - tm />
<cfset tm = getTickCount() />

... code to move, rename, resize...

<cfset times.renamefile = getTickCount() - tm />
```

...you get the idea. What I discovered was that the call to CFDIRECTORY was at least 10 times as expensive as any other operation - sometimes more than 40 times as expensive. Two things contributed to this obvious bottleneck.

When Size Matters

Think about Cfdirectory like you think about Cfquery. If your directory has 10,000 images in it, then you are creating a recordset of 10,000 rows. In our case, a user with 13,000+ images was taking 60 seconds to return a directory list. The other issue is that this site stores the files on a SAN and uses UNC paths to manipulate the files. UNC paths will always perform slower than accessing a local drive.

The Solution

In our case the solution was to stop checking the directory structure and check the database instead. Since the file name was stored in the database under that user, it made sense to simply check and see if that user already had uploaded an image of that name. Another possible solution would be to further segment the directory structure - creating a directory based on the date for example. In any case, in your troubleshooting, don't neglect directories as a possible bottleneck.