Having an image stripped of metadata upon upload in PHP

Asked 15 years, 9 months ago Modified 15 years, 6 months ago

Viewed 5k times Part of PHP Collective

3

A certain site I know recently upgraded their bandwith from 2,5 TB monthly to 3,5 TB.

Reason is they went over the 2,5 limit recently. They're complaining they don't know how to get down the bandwidth usage.

One thing I haven't seen them consider is the fact that JPEG and other images that are displayed on the site(and it is an image-heavy site) can contain metadata. Where the picture was taken and such.

Fact of the matter is, this information is of no importance whatsoever on that site. It's not gonna be used, ever. Yet it's still adding to the bandwidth, since it increases the filesize of every images from a few bytes to a few kilobytes.

On a site that uses up more then 2,5 TB per month, stripping the several thousands images of their metadata will help decrease the bandwidth usage at least by a few Gigabytes per month I think, if not more.

So is there a way to do this in PHP? And also, for the allready existing files, does anybody know a **good** automatic metadata remover? I know of JPEG & PNG Stripper, but that's not very good... Might be usefull for initial cleaning though...

PHP php file-upload metadata vbulletin

Share

Improve this question

Follow

asked Mar 24, 2009 at 11:14

KdgDev

14.5k ● 48 ● 124 ● 158

7 Answers

Sorted by: Highest score (default)

6

It's trivial with GD:

```
$img = imagecreatefromjpeg("myimg.jpg");
imagejpeg($img, "newimg.jpg", $quality);
imagedestroy($img);
```

This won't transfer EXIF data. Don't know how much bandwidth it will actually save, though, but you could use the code above to increase the compression of the images. That would save *a lot* of bandwidth, although it possibly won't be very popular.

Share Improve this answer

Follow

edited Mar 24, 2009 at 11:23

answered Mar 24, 2009 at 11:17

Emil H
40.2k • 11 • 80 • 97

I saw this approach taken on a photography forum, and it was "less than popular" with some users moaning that images were significantly different (as they'd been re-encoded at a consistent quality level) – Rowland Shaw Mar 24, 2009 at 12:25

5

I seriously doubt image metadata is the root of all evil here.

Some questions to take into consideration:

- How is the webserver configured?
- Does it issue http 304 responses properly?
- Isn't there some kind of hand-made caching/streaming of data through php scripting that prevents said data from being cached by the browser? (in which case, url rewriting and http redirections should be considered).

Share Improve this answer

Follow

answered Mar 24, 2009 at 11:22

Julian Aubourg
11.4k • 1 • 30 • 29

It will indeed not be the root of this problem. Yet by eliminating this problem, the site can get some breathing room while taking on the serious issues. When you need to

squeez something out of a tight area, first thing you do is chip bits of the side. – KdgDev Mar 24, 2009 at 12:17

Proper caching may well help a lot more than a bit of meta-data here and there... – Rowland Shaw Mar 24, 2009 at 12:23

@WebDevHobo: suit yourself but beware of unnecessary code bloat for unnecessary fixes ;) – Julian Aubourg Mar 24, 2009 at 13:35

Code bloating you say... I don't think it'll take such proportions. Again, it's not my site, so wheter or not they implement anything is their concern. – KdgDev Mar 24, 2009 at 19:56

well, can their user upload images? In how many places? Do they use pre-made packages that they have no interest hacking in for maintenance reasons? Will you end up having a cron that removes the metadata? There is more to this than meets the eye... but like you said, their call ;)
– Julian Aubourg Mar 24, 2009 at 20:42

3

Check out **Smush.it!** It will strip all un-necs info from an image. They have an **API** you can use to crunch the images.

Note: *By Design*, it ***may*** change the filetype on you. This is on purpose. If another filetype can display the same image with the same quality, with less bytes it will give you a new file.

Share Improve this answer

Follow

answered Mar 24, 2009 at 12:57

scunliffe

63.5k • 26 • 131 • 165

No longer do they have a public API. Their website says that it's in development. – JVE999 Feb 18, 2014 at 3:36

Hmm, that's a bummer about the API. Guess it might be time to work on a tool to do this locally... I don't think this would be too hard with GD just be sure not to "ruin" any JPEG compression when altering the metadata values. – scunliffe Feb 18, 2014 at 14:23

3

I think you need to profile this. You might be right about it saving a few GB but thats relatively little on 2.5TB of bandwidth. You need real data about what is being served most and work on that. If you do find it is images that send your bandwidth usage so high you first should check your caching headers and 304 responses, you also might want to investigate using something like amazon S3 to serve your images. I have managed to reduce bandwidth costs a lot by doing this.

That said, if the EXIF data is really making that much of a difference then you can use the GD library to copy a jpeg image using the imagejpeg function. This won't copy EXIF data.

Share Improve this answer

Follow

answered Mar 24, 2009 at 11:23

Jack Ryan

8,472 • 4 • 41 • 76

2

Emil H's probably addresses the question the best.

But I wanted to add that this will almost certainly not save you as much as you may think. This type of metadata takes up very little space; I would think that

1. Re-compressing the images to a smaller file size, and
2. Cropping or resizing to reduce the resolution of the images

are both going to have a much greater effect. With point one alone you could probably drop bandwidth 50% and with both, you could drop bandwidth 80% - that is if you are willing to sacrifice some image size.

If not, you could always have the default view at a smaller size, with an 'enlarge' link. Most people just browsing will see the smaller image, and only those who want the largest size will click to enlarge it, so you'll still get almost all the bandwidth saving. This is what Flickr does, for example.

Share Improve this answer

Follow

answered Mar 24, 2009 at 11:46

thomasrutter

117k ● 31 ● 152 ● 170

To give a simple example, I once had this program I lined to here take care of a 1,7 GB storage, containing nothing but images. This program will only handle JPG and PNG files, and I still lost 320 MB without loosing any image quality. – KdgDev Mar 24, 2009 at 12:19

I guess if they are all fairly small images then the metadata would take up a greater percentage of their space and you'd save more. I guess I was assuming full resolution photos :) – thomasrutter Mar 24, 2009 at 13:58

Myeah, most these pics are manga drawings, which will rarely go over 1024x768 – KdgDev Mar 24, 2009 at 19:37

0

Maybe some sort of hex data manipulation would help here. I'm facing the same problem and investigating on some sort of automated solution.

Just wondering if that can be done and if possible, I'll write a php class for this.

Share Improve this answer Follow

answered Jun 23, 2009 at 0:11

NiloPC

0

Might be smart to do all the image manipulation on the client side (using a java applet such as facebook does) and then when the image is compressed, resized and fully stripped of unnecessary pixels and content, it can be uploaded at it's optimal size, saving you bandwidth and

server side performance! (at the cost of initial development)

Share Improve this answer

Follow

answered Jun 23, 2009 at 0:16

Shadi Almosri

12k ● 16 ● 60 ● 80