

Re: Intense CPU strain....suggestions

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2007-05/msg02337.html>

- *From:* cbmeeks <cbmeeks@xxxxxxxxxx>
 - *Date:* 31 May 2007 09:45:38 -0700
-

Thanks for the ideas!

I think I might have a somewhat complex system that might work.

First of all, when pictures are uploaded, I record the information in the db. I could add a column that says "is_ready" so that they wouldn't show in the galleries and the resize could mark them ready.

I'm really leaning towards an XML-RPC system. The XML-RPC server could sit and wait for requests. The client ("do_upload") could send a request when uploading is finished. That should take just a second or so...hopefully.

The request could contain information on what to resize.

The XML-RPC server then could instruct the resize script to run and then the script could mark the image ready.

I would need to make sure I knew how to run a robust XML-RPC server capable of handling many requests.

I know this whole system seems complicated but I am trying to think scalability. The XML-RPC server could easily be moved to any machine.

Thanks for suggestions guys.

cbmeeks
<http://www.signaldev.com>

On May 31, 11:40 am, Jerry Stuckle <jstuck...@xxxxxxxxxxxxxxxx> wrote:

cbmeeks wrote:

Well, you could kick off a batch job to do your resizing, then return to the user. That way it can keep processing the images. The only thing

Re: Intense CPU strain....suggestions

you'll have to handle is when the user wants to display his gallery before it's ready.

That's exactly how I want to do it. I just don't know the best way to execute it. I like the idea of having an XML-RPC server running but I am concerned with security (as I should be).

So, let's say I have the following code (using CI):

```
...  
$this->Upload->do_uploads(); // handle the actual uploading.
```

```
$this->Photos->do_resizing(); // handle the resizing
```

```
redirect('waiting');
```

```
....
```

"Waiting" doesn't get kicked off until after do_resizing (sorry, new to PHP). I'm sure there is a smarter way. I'd like to avoid pop-ups, iframes, javascript, etc if possible.

I also thought about having the "do_resizing" simple add some records to a db and have a program running on the server full time that checks the db frequently. But wouldn't that be wasteful? I like the idea of programs laying dormant until something kicks it in the pants and says "do something!" lol

Thanks

cbmeeks
<http://www.signaldev.com>

Re: Intense CPU strain....suggestions

First of all, the redirect won't happen until the code reaches there – which is after your resizing. So that's normal operation.

You could have a cron job kicked off every few minutes to resize the pictures, but again, you'll run into the problem that after uploading the user won't be able to see the pictures he's just uploaded – at least in various sizes.

You could use `shell_exec()` to start off a background process, i.e. (assuming Unix/Linux):

```
shell_exec('nohup myprog file1 file2 file3 2> /dev/null &');
```

`nohup` says to execute a command without hangups and no output, `2> /dev/null` throws away any error messages, and the `'&'` at the end says run in the background.

You'll still run into a problem that the resizing might not occur as fast as the user wants to display it; you'll have to be able to handle this in your display routines.

Alternatively, you could do something like:

```
<body>
echo 'Resizing your images, please wait...<br>';
flush();
$this->Upload->do_uploads(); // handle the actual uploading.
$this->Photos->do_resizing(); // handle the resizing
echo 'Done!. <a href="nextpage.php">Click here to continue</a>';
</body>
```

The `flush()` works with most browsers if you keep the html simple – it forces the output to the browser and the browser displays it. But if you start to get fancy (i.e. tables, etc.), many browsers will wait for the end of the element to actually display.

The only problem with this is, once you've sent *any* output to the browser, you can't redirect them to another page via php (you could use an html redirect). That's why the link.

Just some ideas.

--

=====

Remove the "x" from my email address

Jerry Stuckle

JDS Computer Training Corp.

jstuck...@xxxxxxxxxxxxxx

=====

Re: Intense CPU strain....suggestions