Stable FastCGI configuration on a cPanel server

A default installation of FastCGI on cPanel server is dangerously simple. It's dangerous because one cPanel account (or one vhost) is capable of crashing down a whole server if, say, traffic were to spike up. It's also simple because it won't allow complex scripts to run cleanly. In brief, it's absolutely not ready for production as-is. In this post, I'll go over what it takes to configure FastCGI on a cPanel node properly.

Before you continue reading, be sure to have FastCGI up and running as the PHP handler on your cPanel server. The installation of FastCGI is covered in the online cPanel documentation. From here on now, I'll assume you're ready to add the settings for FastCGI.

The following is a list of settings that you need to add to /etc/httpd/conf/php.conf upon switching to FastCGI:

```
MaxRequestsPerProcess 1000  FcgidMaxProcesses 200  FcgidProcessLifeTim
e 7200  MaxProcessCount 500  FcgidIOTimeout 400  FcgidIdleTimeout 600
 FcgidIdleScanInterval 90  FcgidBusyTimeout 300  FcgidBusyScanInterval
 80  ErrorScanInterval
 3  ZombieScanInterval 3
DefaultMinClassProcessCount 0
DefaultMaxClassProcessCount 3  MaxRequestLen 20468982
```

You're more likely to adjust the settings in **bold** above. *Default**Min**ClassProcessCount 0* instructs FastCGI to keep zero PHP processes running for user when traffic is idle (cPanel account user) . On the other hand, *Default**Max**ClassProcessCount 3* tells FastCGI to never allow more than 3 PHP processes running at a time. This settings prevents one users from crashing the server were they to receive a lot of traffic.

So go ahead and copy/paste the above into your httpd.conf and restart Apache (service httpd restart). You're good to go now!

Please note this configuration will be removed once EasyApache is run next, make sure you have a backup.