

Common Causes of Excessive CPU Usage

This document will cover possible causes of excessive CPU usage. It is a follow up to the Knowledge Base article: [How to interpret resource usage logs](#). If a resource abuse ticket has been opened under your account this information may assist you in resolving the issue.

Most well coded scripts or add-ons/extensions/modules/plugins to scripts will not peak above 35-40% CPU usage. Scripts which cause usage over 70% are generally problematic and, even with light traffic, can cause your site to reach the limit in our shared hosting environment.

Some common causes of high CPU usage are:

- Poorly coded scripts or add-ons/extensions/modules/plugins to scripts.
- Resource intensive scripts such as conversion/generation tools, batch processes and high frequency cron jobs.
- A large number of 'spiders' or 'robots' hitting a website with relatively low 'real' traffic.
- Your website template/design is resource intensive.
- High amounts of 'real' traffic.

Resolving your resource abuse issues

Very often, an out of date script/plugin/component will be contributing to or causing the issue. You will need to update ALL your scripts including any 3rd party add-ons you may have installed. This is critical both to security and performance.

1. Scripts and Add-ons

Many scripts have some built in optimizations. For example, Joomla has caching as a core feature. If your script has this option we would suggest setting a high timeout of at least 120 minutes.

A common misconception is that all websites are the same in terms of CPU/memory usage. This is not the case. For example, a Joomla install with no extensions will use far less resources than an install with 20 active extensions. In addition a Joomla install with 1 poorly coded or resource intensive script can use more resources than a Joomla install with 50 well-coded extensions.

Most developers actively work to improve their scripts as well as patch security holes. Updating to the latest version may help. We would advise keeping the number of extensions you have to a minimum. It is a good idea to uninstall any extensions that you are no longer using.

If high resource usage issues persist after ensuring that all available updates/patches have been applied and unneeded scripts (extensions/components/plugins/add-ons/etc.) have been removed it will be necessary to isolate the problematic script and fix, re-configure, disable or replace it with another script that provides similar functionality. Two common methods are:

Method 1: For Webmaster/Designers

Since PHP debugging/profiling may not be an option due to cost or technical skills requirement, using a process of elimination is typically the quickest way to track down a misbehaving script. To begin, disable all 3rd-party scripts (extensions/components/plugins/add-ons/etc.) within your main script. About 18-24 hours later obtain a new resource usage report to verify that the issue has been resolved.

Next, re-enable 1 or 2 scripts per day and obtain a new resource usage report every 18-24 hours after you make those changes to see if the issue has returned. Once the problematic script has been isolated it will be necessary to take further action to reconfigure the script, disable it, or replace it with a similar script that uses less server resources.

Method 2: For PHP Developers

Profiling of entire PHP pages can reveal high resource usage issues within specific scripts or individual functions. Some scripts have limited profiling features built in but most do not.

This can be done by profiling an offline copy of the site in a development environment along with [Xdebug](#) or similar PHP extensions that have profiling data output capabilities. The most popular offline dev environments are [XAMPP](#), [WAMP](#), and [EasyPHP](#); they are ready-to-run environments which you can use for local testing. XAMPP and WAMP do not include Xdebug by default. However EasyPHP does and also has a preconfigured Webgrind add-on module available.

Once the site has been replicated in an offline dev environment Xdebug should be enabled and configured to record PHP profiling data. There are various frontend applications that can analyze that data and expose any specific functions/scripts that are consuming the majority of the CPU time. Two popular profiling frontends are [Webgrind](#) and [Kcachegrind](#). It will also be possible to see the affect that options within your scripts (such as caching) will have on individual page views when enabled/disabled. Further documentation for Xdebug and profiling frontends as well as details specific to profiling your scripts can be found online.

2. Scheduled Tasks and Batch Processes

The maximum frequency we allow for cron jobs in our shared environment is 1 hour. Please check your cron jobs/batch processes to ensure they are not running too frequently.

3. Template and Design

Using tools such as [Firebug](#) for Mozilla Firefox may help you to troubleshoot and locate issues with your template. The "Net" view in Firebug will display all individual HTTP requests made by your browser. To force your browser to full refresh the page instead of loading content from the cache you'll need to hold down 'Shift' on your keyboard when refreshing the page manually.

Some common problems are:

- Large numbers of scripts and css files
- Large numbers and large sizes of image files being loaded
- Looping RSS feeds (i.e. pulling RSS feeds and feeding them back into the same site)
- Poorly coded templates that are not optimized or resource friendly

Addressing the above, if applicable, will cut down the amount of work the web server has to do for each request. If you have a busy site these changes can add up to a considerable saving of CPU/memory as well as increased site performance.

An excellent GPL (GNU General Public License) tool called [Smartoptimizer](#) can help lower CPU usage and improve site performance by using techniques such as minifying, compression, caching and concatenation of css and js files. Please note that enabling compression/gzip can often make matters worse if no other steps to optimize are taken.

4. Excessive Traffic

Tweaking your robots.txt file may help to cut down on the number of robots/spiders that hit the account. In Awstats under the summary it will show you “viewed traffic” which is real and “not viewed traffic” which is mostly robots/spiders. Details on the [standard robots.txt format](#) can be found online, different robots/spiders will need different tweaks applied though.

Summary					
Reported period	Month Sep 2010				
First visit	01 Sep 2010 - 00:00				
Last visit	15 Sep 2010 - 12:53				
	Unique visitors	Number of visits	Pages	Hits	Bandwidth
Viewed traffic *	916	1261 (1.37 visits/visitor)	7589 (6.01 Pages/Visit)	42745 (33.89 Hits/Visit)	534.69 MB (434.19 KB/Visit)
Not viewed traffic *			498404	605178	29.49 GB

* Not viewed traffic includes traffic generated by robots, worms, or replies with special HTTP status codes.

In the screenshot above of the “Summary” section there is obviously a large difference between the number of pages viewed and not viewed. This is indicating one or more robots are causing problems.

Further down the statistics page you will see the heading “Robots/Spiders visitors” and from here you can see which spider is generating the most hits. Some bots do not identify themselves and will show as Unknown.

Robots/Spiders visitors (Top 25) - Full list - Last visit			
16 different robots*			
	Hits	Bandwidth	Last visit
Unknown robot (identified by 'spider')	540303+529	25.43 GB	15 Sep 2010 - 13:06
Googlebot	30265+31	2.36 GB	15 Sep 2010 - 13:04
Unknown robot (identified by 'crawl')	17715+22	1.54 GB	15 Sep 2010 - 13:05
Unknown robot (identified by 'bot*')	3405+41	131.64 MB	15 Sep 2010 - 13:00
MSNBOT	471+320	17.36 MB	15 Sep 2010 - 13:03
Unknown robot (identified by empty user agent string)	247+18	6.73 MB	15 Sep 2010 - 08:01
Yahoo Slurp	97+28	2.67 MB	15 Sep 2010 - 01:47
Unknown robot (identified by 'robot')	73+14	2.59 MB	13 Sep 2010 - 22:07
MSNBOT-media	27+17	1.03 MB	14 Sep 2010 - 16:40
Ask	23+11	717.78 KB	14 Sep 2010 - 20:44
Unknown robot (identified by hit on 'robots.txt')	0+23	6.83 KB	14 Sep 2010 - 11:25
Voila	5+3	195.90 KB	13 Sep 2010 - 23:35
Alexa (IA Archiver)	1+1	37.19 KB	11 Sep 2010 - 18:30
Unknown robot (identified by '*bot')	1+1	39.30 KB	04 Sep 2010 - 04:41
Netcraft	2	78.00 KB	07 Sep 2010 - 02:15
The web archive (IA Archiver)	0+1	304 Bytes	13 Sep 2010 - 10:50

* Robots shown here gave hits or traffic "not viewed" by visitors, so they are not included in other charts. Numbers after + are successful hits on "robots.txt" files.

Traffic (i.e. users or bots visiting the website) of around 2000 unique visitors per day is typically possible for dynamic sites (For example, Joomla, Drupal or Wordpress based) in our shared hosting environment. Static websites (HTML based) are much lighter on server resources and generally allows for 20-30,000 unique visitors in a day.

Please note there are many factors involved and these figures are an estimate and based on the traffic occurring over a 24 hour period rather than a short space of time.

External statistics such as Google Analytics do not track all traffic. We would advise using Awstats within cPanel as well.

Awstats may also show a single or small number of IP addresses are visiting your site a lot and this may indicate a denial of service attack. IPs can be blocked individually or in ranges via a .htaccess file.

There are [tools](#) available online which can provide further information about IP addresses such as country of origin and what company / organisation they belong to.

5. High Traffic / Busy Website

If so, this is likely a positive thing for you as it means your website is popular and hopefully profitable. If this is the reason your site is abusing resources then you will need to look at upgrading to a more suited platform. Simply open a ticket with our helpdesk and we can explain further our [MVS](#) and [Managed Dedicated Servers](#).

Bandwidth usage on its own is not a good measure of traffic and has little bearing on high CPU usage. High bandwidth usage could simply be due to large files being downloaded rather than many users visiting a website.

Traffic (i.e. users or bots visiting the website) of around 2000 unique visitors per day is typically possible for dynamic sites (For example, Joomla, Drupal or Wordpress based) in our shared hosting environment. Static websites (HTML based) are much lighter on server resources and generally allows for 20-30,000 unique visitors in a day.

Please note there are many factors involved and these figures are an estimate and based on the traffic occurring over a 24 hour period rather than a short space of time.

External statistics such as Google Analytics do not track all traffic. We would advise using Awstats within cPanel as well.