Mozout

Generated on 15/11/2026 00:37

KNOWLEDGE BASE · PUBLIC DOCUMENT

What eats CPU on a website, and how to bring it down

Slow Site & Server Load

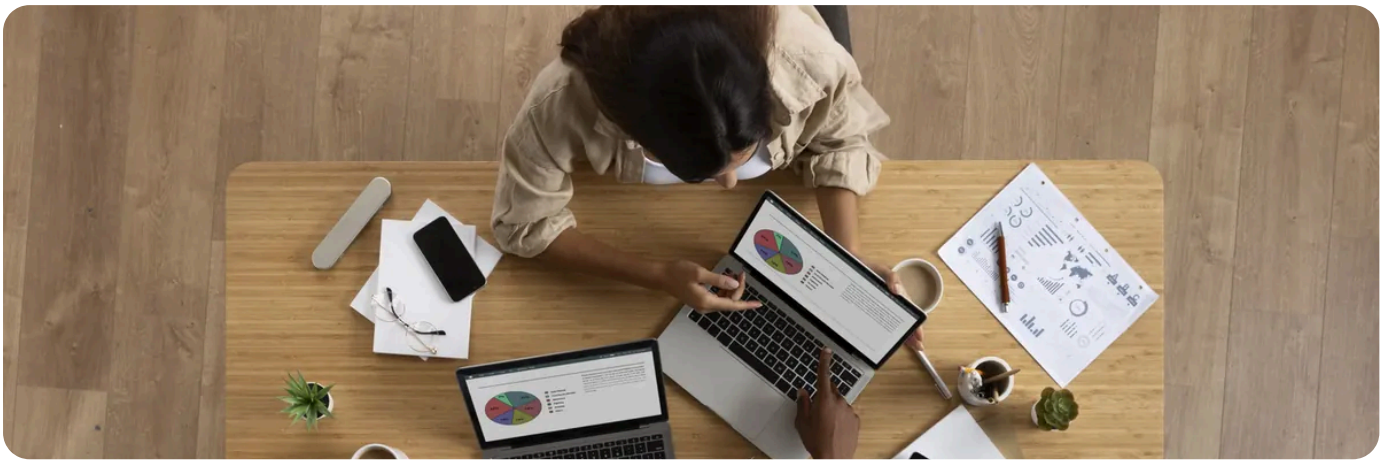
CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-1742	v2026.09.27	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/1742

What eats CPU on a website, and how to bring it down

Slow Site & Server Load Created on **September 27, 2026** 4 min read



The processor is the one resource in your account that does not fail when it runs out: it **slows down**. No error message, no red page, no warning. The site keeps answering, just slowly, and nobody suspects the processor because the processor never complains. That is why this is the limit most people hit without knowing it.

The general mechanics of the limits are in **the limits nobody advertises**, and what to do about a slow site in general is in **the site is slow: what actually makes a difference**. This article is about one thing only: what is burning processor, and how to take the work off it.

What burns processor, heaviest first

The cause	Why, and what you see
Building the page from scratch on every visit	With no cache, every visitor makes the server run the whole program and query the database again, to arrive at exactly the same result. By far the biggest waste, and the easiest to stop.
Heavy database queries	«Related items» lists, internal search, catalogue filters. They show up on specific pages, not across the site.
Bots crawling the site	A bot that walks your whole catalogue every ten minutes burns more processor than all your customers put together, and buys nothing.
Too many scheduled jobs	A job running every minute that should run every hour does sixty times the work. See **cron jobs**.
Plugins that write on every visit	Counters, visit logs, heat maps. Every visit becomes a database write as well.
Processing images on the server	Uploading a huge photograph makes the server generate several versions of it. If the site does that all day, it pays in processor.

The cause	Why, and what you see
Sending e-mail inside the request	A form that sends the e-mail before answering the visitor holds the process open until the mail goes out.

What to do, in the order that pays best

1. **Turn on caching, and the right one.** Our servers run LiteSpeed, and on a WordPress site that means the right plugin is **LiteSpeed Cache**: it stores the finished page inside the server itself, before PHP even starts. A generic caching plugin cannot do that.

2. **Move the site's internal jobs to a real scheduled job.** Many programs, WordPress included, fire their internal tasks off visitor traffic: with little traffic they run late, with a lot they fire constantly. Moving that to a cPanel cron job cuts consumption and makes the timing predictable.

3. **Review how often everything automatic runs.** Open the scheduled job list and ask, one by one, whether that really needs to run that often. Almost none of them do.

4. **Update the PHP version.** The same code runs noticeably faster on a recent version, and faster means less processor per visit. How to do it without breaking anything: **changing your PHP version without breaking the site**.

5. **Hunt down the guilty plugin.** Deactivate half, measure, repeat with the half that remains. Four or five rounds get you there. It is not the number of plugins that counts, it is the weight of each one.

6. **Clean the database.** Revisions of old text, tables from plugins deleted long ago, options loaded on every page. On a database with a few years behind it this is a real gain. With a copy taken first.

One change at a time, with the number written down before and after.
Changing five things at once may well work, but you are left not knowing which one it was, and next time you start from zero. How to measure is in **measuring your site speed**.

Bots, which are half the problem and nobody looks

In a good share of the cases that reach us, most of the requests to the site are not people. They are search engines, price scrapers, model training crawlers and worse. There are three levels of defence, in order of effort.

1. **Look first.** The cPanel statistics show you who asked for most pages, and which ones. See **visitor statistics**. Without looking, you are guessing.

2 **Talk to the polite ones.** A `robots.txt` asking for space between requests is respected by serious search engines, and only by those.

3 **Put a net in front.** A service like Cloudflare filters out what then never touches your processor, and serves images without bothering the server. It is explained in **Cloudflare: when it helps and how to switch it on**.

! **A bigger plan does not cure a badly built site.**
It gives you headroom, not speed. A page the code itself is slow to build takes exactly as long on a bigger machine: what changes is how many of those pages fit at once. If the page is the problem, the bigger plan only postpones the conversation and raises the bill.

When you have cleaned everything and **Resource Usage** still hits the ceiling on real traffic, then yes, the site outgrew the plan. That decision is worked through in **what to measure before you change plan**.

NEED A HAND?

Want us to go through your account's numbers with you? Send us the domain.

Open a support ticket →

SEE ALSO

- **Hosting plans and what each one includes** →
- **VPS server: when the limits become yours to set** →
- **Pricing** →

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-INT-KB-1742 · v2026.09.27 · Generated on 15/11/2026 00:37