interweb

Generated on 22/10/2026 07:28

KNOWLEDGE BASE · PUBLIC DOCUMENT

The site is slow: what actually makes a difference

Speed, Cache & CDN

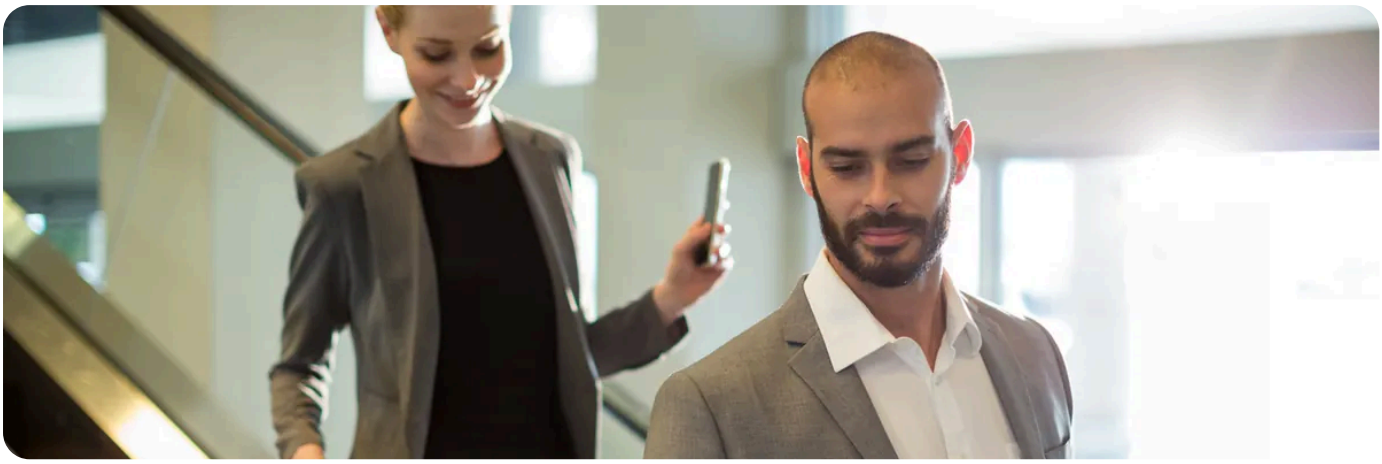
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-292	v2026.09.27	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/292

The site is slow: what actually makes a difference

Speed, Cache & CDN · Created on **September 11, 2026** · Updated on **September 27, 2026** · 8 min read



«The site is slow» means different things depending on who says it: the first page takes ages to appear, the admin panel drags, or a site that used to be fast no longer is. The causes differ and so do the fixes, which is why the first thing to do is not to install anything, but to measure.

Measure first, touch things afterwards

1. Open the site on another network. From your phone on mobile data, for instance. If it is fast there, the problem is your connection or your browser cache, not the site.
2. Try a second page. If only one is slow, the problem is that page, a gallery, a map, a form, and not the server.
3. Notice whether it is only the dashboard. A slow `/wp-admin` with a fast site is nearly always a *plugin*, not a shortage of resources.
4. Look at the consumption in cPanel, under **Resource Usage**: **how to monitor your site performance in cPanel**. If the plan is hitting its ceiling, no amount of image tuning will fix it.

> **Write the numbers down before you start.**
> Change one thing at a time and measure again. Changing five things at once may well work, but you will not know which one it was, and next time you start from scratch.

The short list, in order of payback

If you want the quick version, this is it. Each line is explained below, and the order is the one that gives most result per hour of work.

Order	What to do
1	Move to a current PHP version. Minutes, and the biggest single gain.
2	Turn on the LiteSpeed page cache. Minutes.
3	Deal with the images: size, format and lazy loading.
4	Delete the *plugins* you do not use, and be suspicious of three families.
5	Clean the database and cap the revisions.
6	Only after all that: talk about the plan.

The most common cause here: the PHP version

It is the one we see most and the cheapest to fix. Plenty of sites are still running PHP versions that no longer get fixes at all. A version 8 runs the same code noticeably faster and, unlike the old ones, still gets security fixes.

1. In cPanel, search for **MultiPHP Manager** or **Select PHP Version**.
2. Pick the domain and switch to the highest version 8 the site can take. The step by step is in **changing your PHP version without breaking the site**.
3. Open the site and the dashboard straight after, and click through a few pages. That is the test that counts.

Change it, test it, and know how to go back.
A very old theme or *plugin* may not run on PHP 8, and then the site errors instead of getting faster. It is not serious: you switch back on the same screen. Make the change at a quiet hour.

Caching: our servers run LiteSpeed

Without caching, every visit makes the server build the page again: query the database, run the PHP, assemble it all. With caching, the finished page is served directly. It is the single change that cuts response time most.

Our servers run LiteSpeed. That matters because it means the right *plugin* for WordPress here is **LiteSpeed Cache**: it stores the pages in the server itself, before PHP even starts, something generic caching plugins cannot do.

1. Install the LiteSpeed Cache *plugin* from the WordPress repository.
2. Turn caching on. The default options already give you most of the gain, you do not need to touch everything.

3. After any change to the site, use **Purge All**. If you edited a page and still see the old one, that is the cache doing its job. The full detail, including the exclusions and how to prove it works, is in **installing and tuning a page cache**.

Caching fixes	Caching does NOT fix
Pages that repeat	Pages that differ per visitor: cart, account, checkout. Those have to be excluded.
Traffic spikes	The admin dashboard, which is never cached.
An overloaded server	A page that is slow in itself: if it takes eight seconds to build, the first visit still takes eight.

Never two caching *plugins* at once.
The symptoms are stale pages that will not go away, logins that do not take, and, in a shop, one visitor's cart showing up for another. Pick one and delete the other; deactivating is not enough.

Images: nearly always the biggest single win

The most common reason for a heavy page is not the code: it is photographs uploaded exactly as they came off the camera. A 4 MB image on a page that displays 800 pixels wide makes every visitor download 4 MB to see 800 pixels.

1. Resize before uploading. For most sites 1600 pixels wide is more than enough.
2. Save as WebP where you can: it usually comes out half the size at the same visible quality.
3. Turn on lazy loading, so images load only when the visitor reaches them. LiteSpeed Cache has that option built in.

Plugins and themes

Every active *plugin* costs something on every page, even the ones that do not use it. It is not the number that counts: it is the weight of each one.

1. Deactivate what you do not use, and delete what is deactivated. A deactivated *plugin* does not run, but it is still an unpatched door.
2. Be suspicious of three families: heavy page builders, statistics *plugins* that write every visit into your database, and homepage sliders.

3. To find the culprit: deactivate half, measure, and repeat with the half that remains. Four or five rounds get you there.

When it is only the dashboard that drags

A fast site with a slow dashboard is a different problem, and caching does not solve it, because the dashboard is never cached. The causes are nearly always these four:

Cause	What you notice, and what to do
The editor heartbeat	While you are writing a page, WordPress talks to the server every few seconds to autosave and to flag that the page is being edited. In a tab left open all day, that is hundreds of requests. There are *plugins* that reduce how often it happens.
`admin-ajax.php`	If you see this file repeating in the access log, a *plugin* is talking to the server non-stop, often on the public side of the site too: **admin-ajax eating your resources**.
The internal scheduler	WordPress runs scheduled tasks on the back of visits. On a site with few visits they pile up and all fire on top of the next visitor. The cure is **swapping WP-Cron for a real server cron job**; the general context is in **cron jobs**.
Dashboard widgets	News boxes, statistics and *plugin* notices that fetch data from outside servers. Hide what you do not read, in the **Screen Options** menu at the corner.

The database

Over the years invisible rubbish piles up: a revision of every text ever edited, data from *plugins* removed long ago, and options loaded on every page. On a site with a few years behind it, clearing this out gives a real gain.

1. Take a backup first: **how to restore your data**.
2. Cap future revisions with `define( 'WP_POST_REVISIONS', 5 );` in `wp-config.php`. It keeps the last five of each text instead of every one ever made.
3. Empty the bin sooner with `define( 'EMPTY_TRASH_DAYS', 7 );`.
4. To clear out what is already there, command by command: **cleaning and optimising the database in phpMyAdmin**.

One thing weighs on EVERY page.
The options table has rows marked to load on every single request. A badly written *plugin*, or one you already deleted, leaves enormous rows there that go on being read a thousand times a day. It is worth looking at the biggest ones before looking at anything else.

What comes from outside your house

Part of the time the visitor waits is not spent on our server: it is spent fetching things from other people. External fonts, embedded maps, videos, share buttons, chat widgets and analytics tags. Each one is a fresh connection to a server you do not control, and if that one is slow, your site looks slow.

Count what you have, and remove what is serving nothing. Fonts can be served from your own site; videos can load only when somebody presses play.

When it is not the site, it is the plan

There is a point past which optimising stops paying off. If **Resource Usage** shows the limit being hit every day, or if error 508 appears, the site no longer fits the plan: **website errors: 500, 403, 404 and 508** and **the limits nobody advertises**. What to measure before deciding is in **the site is slow: what to measure before you change plan**, and the honest comparison in **shared hosting vs VPS**.

Careful with aggressive «optimisers».
Options like combining and minifying all the CSS and JavaScript, or deferring everything, buy tenths of a second and break menus, forms and carts with impressive ease. Turn one on at a time and use the site properly after each, including buying something, if you sell. If something breaks: **common WordPress errors**.

NEED A HAND?

Want us to look at what is weighing your site down?

Talk to us →

SEE ALSO

- Hosting plans and what each one includes →
- WordPress hosting →
- Support Policy →

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-292 · v2026.09.27 · Generated on 22/10/2026 07:28