FIELD NOTES FROM A PRESALES SOLUTIONS ARCHITECT

Rank Math Sitemap Not Updating — the Cache That Was Not LiteSpeed or Cloudflare

A field note about a sitemap that froze for almost three months on my own site — every new post missing, Google reading an old file — and the header check that told me to stop purging caches and look at the plugin.

Daanial Farrukh · 27 November 2026 · https://quabin.com/rank-math-sitemap-not-updating-stale-cache/

This one did not happen in a customer room. It happened on this blog.

I was reading the Search Console report for quabin.com at the end of August. One of my articles, a cPanel mail server checklist published in late June, sat in the Page indexing report under "Crawled - currently not indexed".

I opened the sitemap to check it was listed.

It was not.

The sitemap held **2 URLs**. The site had more posts than that. And Search Console said it had last read the sitemap on **9 June 2026**. Nearly three months of nothing.

My first thought was the one everyone has on a WordPress site behind a page cache and a CDN.

"It's cached somewhere. Purge LiteSpeed. Purge Cloudflare."

That would have been the wrong afternoon.

That's the trap. A stale sitemap looks exactly like a caching problem. It is the first thing every guide tells you to purge. But when the cache is inside the SEO plugin itself, no page cache purge in the world will touch it.

What a Sitemap Cache Actually Is

Rank Math builds the XML sitemap on request, from the posts in the database. Building it every time is wasteful, so it keeps a ready copy.

A sitemap cache is a stored copy of the generated sitemap that the plugin serves until something tells it the content changed.

Publishing a post is supposed to be that signal. The flow is meant to be:

Publish post → Rank Math clears its sitemap cache → next request rebuilds the sitemap → Google reads the new file

On my site the second step stopped happening. Rank Math kept serving the copy it had built before 9 June. Every post published after that date was simply not in the file.

There are three layers on a typical WordPress stack that can hold an old sitemap, and they look the same from outside:

- **The CDN** — Cloudflare or similar, caching the XML at the edge.
- **The page cache** — LiteSpeed Cache, WP Rocket, Varnish, caching the response on the server.
- **The application** — the SEO plugin's own stored copy in the WordPress database.

Only the first two are what people mean by "the cache". The third one is the one nobody purges.

Why Purging Everything Did Not Look Wrong

Here's the part that trips people up.

On most days, a page cache purge does fix a stale page. So it becomes a reflex. You purge, you reload, you see the same old file, and you assume the purge did not reach the right edge. So you purge again. Then you add the sitemap to the page cache's exclude list. Then you start reading about Cloudflare Cache Rules.

Every one of those steps is reasonable. None of them can change what the application generates.

And the symptom in Search Console pointed the wrong way too. "Crawled - currently not indexed" sounds like a quality judgement on the article. It made me reread the post before I looked at the plumbing.

A frozen sitemap does not throw an error. It just quietly stops telling Google about anything new.

How to Actually Find Which Layer Is Stale

The rule I follow now: before purging anything, ask each layer whether it served the response.

Step 1: Fetch the sitemap with a cache-buster and read the headers

A random query string makes most caches treat the request as new.

```
curl -sI "https://example.com/post-sitemap.xml?nocache=12345"
```

You are reading three headers.

cf-cache-status tells you what Cloudflare did. DYNAMIC means Cloudflare did not cache this response at all; it went to the origin. HIT means the edge answered from its own copy.

x-litespeed-cache-control (or your page cache's equivalent) tells you whether the server-side cache stored it. no-cache means it did not.

And x-robots-tag: noindex will very likely be there too. That one is normal on Rank Math sitemaps. Google still reads the file. Leave it alone.

On my site the answer was cf-cache-status: DYNAMIC and x-litespeed-cache-control: no-cache. Both outer caches had stepped aside. The response came fresh from WordPress.

Step 2: Read the body that came back fresh

```
curl -s "https://example.com/post-sitemap.xml?nocache=12345" | grep -c "<loc>"
```

This counts the URLs in the file. Compare it with the number of published posts you expect.

Mine said 2. The fresh, uncached, straight-from-the-app body was still wrong.

This is the verdict line: no outer cache served it, and the body is still wrong, so the application is generating the wrong output. Stop chasing caches.

Step 3: Look for a clean cut-off date

```
curl -s "https://example.com/post-sitemap.xml?nocache=12345" | grep "<lastmod>"
```

Find the newest lastmod in the file. Then list your posts published after that date.

If every missing post is newer than the newest lastmod, and every older post is present, you are looking at a snapshot. Something built the file on that date and has been serving it ever since. That clean cut-off is the fingerprint of an application-level cache, not a random missing entry.

On my site every missing post was newer than the file. The checklist from late June was the clearest one.

How to Fix It

Step 4: Flush Rank Math's sitemap cache

The gentle way first:

Rank Math SEO › Sitemap Settings › **Save Changes**

Change nothing. Just save. Saving the sitemap settings flushes the stored sitemap, and the next request rebuilds it from the database.

If that does not clear it, there is a direct tool. In Easy Mode Rank Math hides it, so:

Rank Math SEO › Dashboard › switch to **Advanced Mode**, then Rank Math SEO › Status & Tools › Database Tools › **Delete Sitemap Cache**

On my site the Save Changes click was enough.

Step 5: Verify the file, not the button

This step is where most tutorials stop and where mine starts.

Run Step 2 again with a new cache-buster:

```
curl -s "https://example.com/post-sitemap.xml?nocache=67890" | grep -c "<loc>"
```

The count should now match your published posts. Mine went from 2 to 4, and the missing checklist article was back in the file. Then check that your newest post is in it:

```
curl -s "https://example.com/post-sitemap.xml?nocache=67890" | grep "your-newest-slug"
```

If it prints nothing, the rebuild did not happen. That is not a failure of the check. It is the check doing its job.

Step 6: Tell Google

Search Console › Sitemaps › enter the sitemap index URL › **Submit**. The "Last read" date should move forward within a day or so.

Then, for each post that was missing, use URL Inspection and click **Request Indexing**. I did this for the checklist article the same day.

One more thing I learned here. URL Inspection told me that article was already on Google and the page was indexed — while the Page indexing report still listed it as not indexed. The report lags by roughly a week. **For a single URL, trust URL Inspection over the report.**

What I Kept, and What Was Not the Fix

During the hunt I added sitemap to LiteSpeed Cache › Excludes › Do Not Cache URIs. That is a correct setting to keep. A sitemap should never be page-cached, because the page cache would then add a second layer of staleness on top of the plugin's.

But it was not the fix. The headers had already shown LiteSpeed was not serving the file. I want to be honest about that, because it would be easy to write this up as "exclude the sitemap from LiteSpeed" and send the next person down the same wrong road.

The Checklist I Run Now

1. **Count the URLs.** curl -s "<sitemap>?nocache=N" | grep -c "<loc>" against the number of published posts.
2. **Is the newest post listed?** grep the newest slug in the same output.
3. **Who served it?** curl -sI with a cache-buster and read cf-cache-status and your page-cache header.
4. **Is there a cut-off?** Newest lastmod in the file versus your newest post date.
5. **Flush at the right layer.** Fresh headers plus wrong body = plugin cache. Rank Math › Sitemap Settings › Save Changes.
6. **When did Google last read it?** Search Console › Sitemaps › Last read. Weeks old on an active site is a warning.

The whole list takes two minutes. I now check item 2 after every post, as part of publishing.

What I Tell Every Customer Now

When a client says "Google is not picking up our new pages", I ask one question before anything else:

"Is your newest page in the sitemap that the server is generating right now — and how do you know it wasn't served from a cache?"

If the answer is a curl with fresh headers and the slug in the output, they're in good shape. If the answer is "we purged the cache", we're not done yet.

I write about the questions I ask in real customer rooms — the ones that change architectures and surface the problems no one else noticed. If you found this useful, drop your email below and I'll send the next one when it's ready.

FAQs

Why is my Rank Math sitemap not updating with new posts?

Rank Math keeps a stored copy of the generated sitemap and serves it until the cache is cleared. If that copy stops being cleared on publish, new posts never appear. Go to Rank Math SEO › Sitemap Settings and click Save Changes without changing anything to flush it, then check the sitemap again with a cache-buster.

How do I know if Cloudflare or LiteSpeed is caching my sitemap?

Fetch the sitemap with a random query string using curl -sI and read the headers. cf-cache-status: DYNAMIC means Cloudflare passed it to the origin, and x-litespeed-cache-control: no-cache means LiteSpeed did not store it. If both say that and the body is still old, the stale copy is in the application, not in those caches.

Where is the Delete Sitemap Cache button in Rank Math?

It is under Rank Math SEO › Status & Tools › Database Tools. Rank Math hides it in Easy Mode, so switch the plugin to Advanced Mode on its dashboard first. Try Sitemap Settings › Save Changes before this, because it is the gentler step and is often enough.

Is the x-robots-tag noindex header on my sitemap a problem?

No. Rank Math sends x-robots-tag: noindex on its sitemap files so the XML itself does not appear in search results. Google still reads the sitemap and follows the URLs in it. Leave the header as it is.

Should I exclude the sitemap from my page cache?

Yes, it is good practice, because a page-cached sitemap adds a second layer that can go stale. In LiteSpeed Cache the setting is Excludes › Do Not Cache URIs. But excluding it will not fix a sitemap that the SEO plugin itself is serving from its own stale copy.

The Pages report says my post is not indexed, but URL Inspection says it is on Google. Which is right?

URL Inspection shows the current state of that one URL. The Page indexing report is rebuilt in batches and can lag by about a week. Trust URL Inspection for a single URL, and expect the report to catch up later.

Read online: https://quabin.com/rank-math-sitemap-not-updating-stale-cache/