



Toolforge webservices are in the final stages of [migrating to the toolforge.org domain](#) .
Please help us clean up older documentation referring to tools.wmflabs.org!

Incident documentation/20140313-API-Parsoid

[< Incident documentation](#)

Summary

From Faidon's email to the Ops mailing list

At 06:19 UTC, Erik dropped in to #wikimedia-operations after noticing extreme Parsoid slowness, up to 40s for relatively simple pages. Erik immediately paged Gabriel & Roan (which joined began to investigate) and at 07:11 he also paged me and I joined the fun at 07:28.

After a brief moment where Search was blamed (because it's broken, see <http://gdash.wikimedia.org/dashboards/searchlatency/> and spikes of 500ms latency at 06:20, probably index rebuilding or something), the initial investigation showed API appservers overload, reaching up to 100% at times. Roan had already attempted to address that by rebalancing the API appserver pool weights but I pointed out that the original weights were correct because a) the "24-core" servers are actually 12-core with HT, b) the weights are adjusted based on memory, not CPU. We quickly reverted that.

Both Roan and me continued down that rabbithole and we both found out via different avenues that there was no real req/s spike but there was one user, coming from 5 different IPs (<redacted>) URLs such as:

<http://el.wikipedia.org/w/api.php?<redacted>> (all elwiki, all action=parse with the same prop, different oldid)

The logs show them taking from 40-100s to complete and we had plenty in the sampled 1:1000 logs that correlate exactly with the CPU spikes we were seeing, so there's a high probability that these were the trigger.

However, the CPU spike ended at about 07:25 and Erik & Gabriel were still seeing occasional Parsoid requests taking ~16s for small pages, which was also confirmed by <http://gdash.wikimedia.org/dashboards/ve/> & <http://gdash.wikimedia.org/dashboards/apimethods/> (MediaWiki.API.visualeditor, which curls Parsoid, which curls the API -- fun!). Moreover, Gabriel was reporting that hitting the frontend Varnishes via a local Parsoid instance instead of directly hitting the API LVS was working correctly. The graphs also showed that request latency spiked as early as 13:00 UTC and hence it clearly preexisted of the CPU spike.

At that point, various minor points I had seen in Ganglia & SAL struck me and at 08:30 I identified and subsequently fixed the issue. The issue was that Chris, per RT #7000, had moved three appservers of the API pool (mw1021-mw1023) to row D to offload two overloaded racks. The move required an IP renumbering (& associated DNS change) due to eqiad's network layout. The problem was that Pybal didn't cope well with that, which in turn meant that it kept the *old*, now unresponsive, IPs of those three servers in round-robin LVS rotation. A small portion of requests were thus reaching those three servers, which meant timeouts. Both Varnish & Parsoid retry on failures, however Varnish's connect timeout is 5s (set from cache.pp), while presumably Parsoid's is 15s which explains the 16s occasional latency.

I removed the old IPs manually using ipvsadm and restart pybal on both active & backup LVS (lvs1003 & lvs1006) and graphs immediately confirmed that this was the root cause and right fix.

Actionables

- Status: ■ **informational** - Search is broken, with latency quadrupling to crazy numbers on a daily basis. We kinda knew that :(I'll leave the decision of what to do (fix or wait for ElasticSearch) to Nik.
- Status: ■ **Declined** - Investigate throttling of the API, address DOS vectors
 - https://bugzilla.wikimedia.org/show_bug.cgi?id=62615 (private bug)
- Status: ■ **Declined** - The Pybal DNS bug needs to be fixed. Until then remove servers from /h/w/conf/pybal before renumbering them or decommission them. It's probably a good idea anyway.
 - <https://rt.wikimedia.org/Ticket/Display.html?id=7127>
- Status: ■ **Declined** - Make Pybal better about error detection/logging (it hides/makes opaque some backend errors)
 - <https://rt.wikimedia.org/Ticket/Display.html?id=7152>

[Main page](#)
[Recent changes](#)
[Server admin log \(Prod\)](#)
[Server admin log \(RelEng\)](#)
[Deployments](#)
[SRE/Operations Help](#)
[Incident status](#)

[Cloud VPS & Toolforge](#)

[Cloud VPS documentation](#)

[Toolforge documentation](#)

[Request Cloud VPS project](#)

[Server admin log \(Cloud VPS\)](#)

[Tools](#)

[What links here](#)

[Related changes](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)






[Cite this page](#)

[Print/export](#)

[Create a book](#)

[Download as PDF](#)

[Printable version](#)

- It's unfortunate that we noticed such an issue hours later via a user report. We should have an alert for unusual/high API latency (among others). The data is there, in Graphite, but we need a check_graphite to poll it. Matanya started that but it needs more work.
 - Status: ■ **On hold** - <https://gerrit.wikimedia.org/r/#/c/118435/> 
 - Status: ■ **Declined** - https://bugzilla.wikimedia.org/show_bug.cgi?id=57882 
 - Status: ■ **Declined** - Talk with Brad on his thoughts on monitoring the API (error rates etc) - **NEEDS BUG**
- Similarly, we probably need to monitor reports for failing/retried requests & alert when they happen. The current reqstats/reqerror graphs report errors from frontends which in this case were showing no errors as they were retrying and succeeding. We really need to overhaul the whole metric collection & alerting there.
 - Status: ■ **Declined** - [T83580](#) 
 - Status: ■ **Declined** - Talk with Analytics team, they probably have related analytics
 - Status: ■ **informational** - Good candidate for a monitoring Sprint at the Hackathon
- Status: ■ **Done** - Reconcile the Parsoid/Varnish connect/TTFB timeout differences.
 - Gabriel votes that Varnish's 5s is more sensible than Parsoid's 15s.
 - [\[1\]](#)  and [\[2\]](#) 

Category: [Incident documentation](#)

This page was last edited on 1 June 2015, at 12:58.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

[Privacy policy](#) [About](#)
[Wikitech](#)

[Disclaimers](#) [Code of Conduct](#) [Developers](#) [Statistics](#) [Cookie statement](#) [Mobile view](#)

