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

Incident documentation/20200511-thumbor

< Incident documentation

document status: draft

Summary

An external client issued a really large number of requests for what appears to be nonexistent original images uploaded to commons that caused the thumbor service to return 503s and eventually paged.

Impact: Approximately 291,000 thumb-nailing/resizing for images (which is what the thumbor service is for) requests failed and were returned as 503s (an HTTP error code). 76,000 of those from the offending IP. The incident affected eqiad and esams, so predominantly Europe and East Coast of the American Continents.

Contents [hide]

Timeline

All timelines are on 2020-05-11 and are UTC

We were paged by icinga, roughly 40 mins after the behavior started.

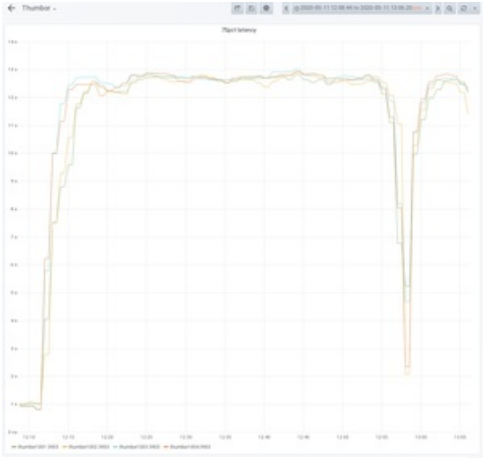
- 12:11 approximately: Thumbor is starting to return a large number of 404s and a little bit later of 503s. Peaks are at 75rps and 104 rps respectively. Incident begins. It will take another 36 minutes before it becomes critical
- 12:47: SRE gates paged. Multiple people respond
- ~12:50: It is noted in graphs that thumbor is serving a lot of 404s and 503s[1] to the caching proxies. Latencies have skyrocketed. From 1s to 12.5s for the p75[2] and from 5s to 15s[3] for the p98
- 12:54: A single IP gets noticed for having requested 20x the number of requests the 2nd in order has.
- 13:02: A rule is put in the caching proxies to block the aforementioned IP
- 13:08: It becomes apparent that block isn't working.
- 13:13: The block is fixed and set correctly. However the block is for the IPv6 only (as the IPv4 one isn't known)
- 13:18: It becomes apparent that swift is setting Cache-control: no-cache when returning 404s which disallows caching, even for a short amount of time on the caching proxies, which would have worked as a back pressure mechanism
- 13:19: The offender falls back to their IPv4 address now that their IPv6 is banned.
- 13:23: Questions on why Thumbor's rate-limit for originals isn't kicking in
- 13:26: Blocks are updated with the IPv4 address. This time around the solution seems to hold
- 13:26: Incident ends
- 13:27: Question about the haproxy queue not having anything in it. The premise of the queue is to buffer requests when thumbor is under stress, which it did not do for some reason

Links to dashboards

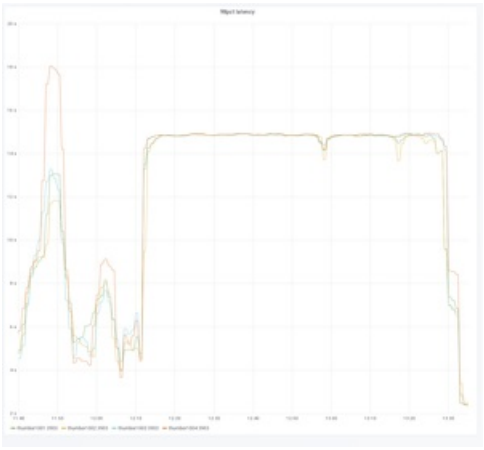
- [1] https://grafana.wikimedia.org/d/Pukjw6cWk/thumbor?panelId=39&fullscreen&orgId=1&from=1589198924772&to=1589202380033
- [2] https://grafana.wikimedia.org/d/Pukjw6cWk/thumbor?panelId=35&fullscreen&orgId=1&from=1589198924772&to=1589202380033
- [3] https://grafana.wikimedia.org/d/Pukjw6cWk/thumbor?panelId=34&fullscreen&orgId=1&from=1589198924772&to=1589202380033

Screenshots of the above dashboards are below as well:

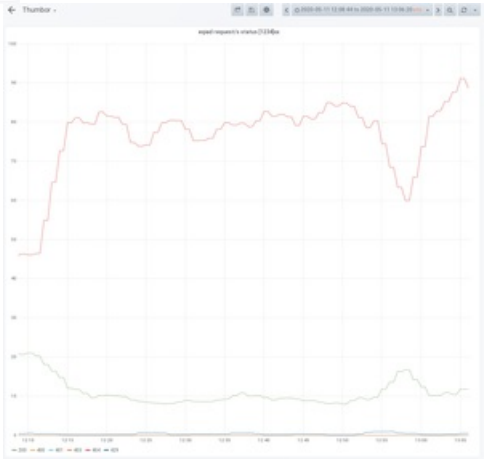
P75 latency:



P98 latency:



Error rate:



Detection

We were paged by icinga, roughly 40 mins after the behavior started. The alert was the LVS one, about the entirety of the service. No alerts prior to that, despite the event having started 30+ minutes earlier.

Conclusions

The current per-IP throttling implemented in Thumbor is inadequate. It attempts to create a system where a given IP can only use X workers and have a queue of Y, but since it's based on PoolCounters requests beyond the X limit are actually served by other Thumbor workers. We end up with a situation where the client can keep a lot of workers busy… waiting on its own throttling lock. This wasn't the intention and as such the PoolCounter-based throttle is broken.

What went well?

We were quickly able to identify the cause of the incident and react to it. Automated monitoring detected the incident, albeit relatively late.

What went poorly?

Thumbor as a service was under stress for ~40m before it paged. No one noticed before that. Blocking of the offender took more than ideal due to (in this order), operator error, IPv6 => IPv4 fallback, unrelated bad puppet deploy.

Where did we get lucky?

It happened at a good time for most of the SRE people, performance team was also present in the IRC channels, we were able to diagnose and block the offending IP.

How many people were involved in the remediation?

Multiple SREs, 1 SRE as incident coordinator and 2 people from performance.

Links to relevant documentation

https://github.com/wikimedia/puppet/commit/9b4dde717c7106d72dadf743d2b3d3eb70ed4f1c change to stop unconditionally caching 404s in the caching layer that exacerbate the problem

Actionables

- Add a Cache-Control header to 404 responses coming from Thumbor/Swift Proxy if there isn't one yet https://phabricator.wikimedia.org/T252425
- Consider a very short term cache (5-10 min?) of 404's for thumbnails, bearing in mind the possibility for cache pollution attacks
- Define who is in charge of basic Thumbor maintenance.
- Consider adding alerting for Thumbor query success rate, or for p50/p75 latency.
- Lower poolcounter per-IP limits in Thumbor as much as possible while not breaking the Commons new uploads page too much. https://phabricator.wikimedia.org/T252426

Categories: Incident documentation | Incident documentation drafts