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

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/20200204-maps

< Incident documentation

document status: in-review

Contents [hide]

Summary

Maps servers fully saturated on CPU, resulting in an increase in user-experienced latency and request error rate. In order to shed load and restore service for users of Wikimedia projects, traffic from non-Wikimedia sites was blocked (and as of 2020-03-04, is still blocked).

The proximate cause is not fully known, but a large part of it is likely fallout of a Mediawiki API outage, as Maps servers need to fetch data from the Mediawiki API for some kinds of requests, and there have been previous Mediawiki incidents where similar Maps impact was seen.

The deeper causes involve a long-running lack of staffing on the software and its infrastructure, one of the manifestations of which is a lack of familiarity with the software and the infrastructure by the SRE team and most others involved in day-to-day operations.

Impact

All Maps users, including those of Wikimedia wikis and projects, were experiencing elevated latency and error rates from 16:03 until 21:20.

All Maps users, including those of Wikimedia wikis and projects, were unable to load tiles that were not already cached by our CDN, for approx 20 minutes from 21:00 to 21:20.

"External" Maps users (those on non-Wikimedia wikis and projects – everything from travel agencies to Pokémon GO fan sites) are unable to load tiles that were not cached by our CDN starting at 21:00 and ongoing as of this writing.

About 1.44M errors were served to users for the duration of the outage. To mitigate the outage, we disabled external traffic that accounts for about 36% of requests to Maps.

Detection

Automated detection via basic LVS/PyBal alerts.

However, note that users were already experiencing elevated latency and errors well before the extant alerts fired.

Timeline

All times in UTC.

See also: Incident documentation/20200204-app server latency

- 15:38: Mediawiki appservers begin to slow down.
- 16:03: Large spike of appserver errors and latency. Maps begins serving a low rate of HTTP 503 and 504 errors, starting around 4rps and increasing to 50rps. **OUTAGE BEGINS**
  - Kartotherian latency almost certainly increases as well, but there's no trusted monitoring on this.
- 16:09: Maps CPU usage in eqiad, already very hot under normal load at 70%+, skyrockets to 100%. [1]
- 16:12: Appservers return to normal latency and error rate.
- 16:19: First alert related to Kartotherian/Maps: PROBLEM - PyBal backends health check on lvs1016 is CRITICAL: PYBAL CRITICAL - CRITICAL - kartotherian-ssl_443: Servers maps1004.eqiad.wmnet are marked down but pooled https://wikitech.wikimedia.org/wiki/PyBal
  - This alert fires, then recovers, for several of the Maps servers. This alert indicates that Maps servers are beginning to become so overloaded that they cannot reply to PyBal's healthchecking queries in a timely manner.
- 16:26: First page related to Kartotherian/Maps: PROBLEM - Kartotherian LVS eqiad #page on kartotherian.svc.eqiad.wmnet is CRITICAL: /v4/marker/pin-m-fuel+ffffff@2x.png (scaled pushpin marker with an icon) timed out before a response was received: /v4/marker/pin-m-fuel+ffffff.png (Untitled test) timed out before a response was received https://wikitech.wikimedia.org/wiki/Maps%23Kartotherian
  - This alert indicates that zero Maps@eqiad servers are healthy enough to respond to PyBal's healthchecking queries in a timely manner.
- 16:45: akosiaris stops all kartotherian processes on maps servers, seeing an increase in requests to Maps prior to the start of 20200204-app server latency – which leads to a belief that perhaps traffic from Maps to Mediawiki is the cause of the API server outage.
- 16:45: In addition to the lower rate of HTTP 503 and 504 errors, users begin experiencing many HTTP 502 errors. Maps is serving approx 300rps of errors, or about 17% of Maps traffic. [2]
- 17:41: akosiaris restarts the Karotherian service on maps100*. CPU consumption quickly returns to 100%.
- 17:41: HTTP 502 errors end.
- 18:45: cdanis, believing the problem maybe has something to do with eqiad specifically, and not realizing that Maps is in a global capacity crunch, depools Maps@eqiad, forcing all load over to codfw -- which then also begins suffering from 100% cpu consumption.
- 19:43: cdanis blocks Maps requests with a Referer headers matching twpkinfo.com, a Pokémon GO fan site, which was approx 25% of maps load at the time https://gerrit.wikimedia.org/r/c/operations/puppet/+/570129
- 20:31 shdubsh restarts Kartotherian on maps2001 after enabling extra logging for analysis
- 21:00: akosiaris modifies our CDN configuration to block all *cache misses* that are from non-WMF Referers: 570140
  - As it is only CDN cache misses that are forwarded to the Maps servers and cause extra load, this allows "free" tiles (from a Maps CPU perspective) to continue loading on external sites, minimizing the impact while preserving the service
  - However, this change is actually subtly erroneous, and despite multiple reviewers, we block all Maps cache misses, WMF site or not.
- 21:20: cdanis deploys 570143 which allows akosiaris's change to work correctly. **OUTAGE ENDS**
  - Maps CPU usage in codfw returns to about 80% instead of 100% or 0%. [3]
- 21:49: cdanis deploys 570147 which allows wmflabs.org sub-domains (forgotten in the original change) to continue to use Maps, as well as fixing some other edge cases.
- 22:03 cdanis repools Maps@eqiad.
  - CPU usage in codfw drops to about 17% [4] and CPU usage in eqiad rises to about 40% [5]

Conclusions

What weaknesses did we learn about and how can we address them?

The following sub-sections should have a couple brief bullet points each.

What went well?

- Many SREs were around to work on the incident.

What went poorly?

- Maps issues blocked a Mediawiki train deploy, incurring lost productivity for many, across many teams and even the volunteer technical community.
- We accidentally blocked all Maps traffic, including Wikimedia-originated traffic we intended to allow, for a

period of ~20 minutes.
- We removed a bunch of maps capacity (maps@eqiad) and left it offline for a long time, not understanding that maps was **both** active/active **and** already globally underprovisioned.
- There is a recurring problem where, when Mediawiki API servers suffer from elevated latency/error rate/other overload symptoms, Kartotherian backend CPU explodes and stays that way for an indeterminate period of time. (For the corresponding Mediawiki incident here, see 20200204-app server latency. For a previous occurrence of this, see 20200126-app server latency.)
  - There are insufficient developer resources and no SRE resources assigned to investigate this issue.
- Kartotherian's monitoring dashboard is opaque to non-experts, and there are very few Kartotherian experts.
  - For example, there are no units indicated on the 'performance' graph, which seems to measure (probably) latency in (possibly) milliseconds?
  - There are insufficient developer resources and no SRE resources assigned to fix these issues.
- There's very little familiarity with the internals of the Maps service amongst most of the SRE team.

Where did we get lucky?

- *for example: user's error report was exceptionally detailed, incident occurred when the most people were online to assist, etc*

How many people were involved in the remediation?

At least 6 SRE for 6 hours.

Links to relevant documentation

Where is the documentation that someone responding to this alert should have (runbook, plus supporting docs). If that documentation does not exist, there should be an action item to create it.

Actionables

Explicit next steps to prevent this from happening again as much as possible, with Phabricator tasks linked for every step.

NOTE: Please add the #wikimedia-incident Phabricator project to these follow-up tasks and move them to the "follow-up/actionable" column.

- Rebalance Maps traffic between eqiad and codfw: eqiad is often much hotter than codfw, as it sees all the European load. (TODO: Create task)
  - Making more use of codfw (and ulsfo) in general is desired by the Traffic team; can potentially use Maps as a proving ground for a new geographical DNS mapping.
- Make a policy decision about whether or not to continue disallowing external sites to embed our maps. If we decide to do so:
  - Probably, begin disallowing cache hits as well as cache misses; it's confusing.
  - Announce this publicly; remove Wikimedia from OSM's list of map tile providers; etc.
- Perform some capacity planning for Maps. phab:T228497
  - This was an actionable in a few previous Maps outages, but probably didn't happen?
    - Incident documentation/20190308-maps
    - Incident documentation/20190715-maps
    - Incident documentation/20190913-maps
- Investigate why Maps overloads when appservers have high latency (or are returning errors)
  - One of the Kartotherian log messages that was prominent in the incident was `Bad geojson - unknown type object` which seems to correlate with appserver trouble.
- Attempt to get some more staffing behind Maps?

Categories: Incident documentation in-reviews | Incident documentation | Maps outages, 2020

This page was last edited on 4 March 2020, at 15:05.