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/20200522-thumbnails

< Incident documentation

document status: in-review

Summary

A change was made to the Memcached configuration of Wikimedia Commons. The change would distribute cache keys in a more optimal way (T252564). It had the expected impact that there would be a temporarily increase in re-computations, and therefore it was applied to only one big wiki at a time. Each wiki has its own cache namespace, and there is a global namespace for shared cache keys. Global keys must be distributed in the same way across all wikis (regardless of the wikis' own cache configuration), to avoid a split-brain scenario.

It turned out that for many years, the cache keys relating to thumbnail metadata were wrongly marked as "local" instead of "global". Which meant that Commons' now used the new distribution, while Wikipedia used the old still, thus a split-brain scenario.

From first report to resolution took 4 hours. Total time the issue is presumed to have existed is 21 hours.

Impact: About 10% of media uploads hit a race condition causing them to appear to not exist and thus unable to be inserted into articles. No data was lost. All affected files eventually became accessible either by themselves after a cache churn, or when our fix was deployed.

Timeline

All times in UTC.

Week of 11 May 2020

- Tue 2020-05-12 19:05 <hashar@deploy1001> synchronized wikiversions files: group0 wikis to 1.35.0-wmf.32
- Wed 2020-05-13 19:08 <hashar@deploy1001> synchronized wikiversions files: group1 wikis to 1.35.0-wmf.32
- Wed 2020-05-13 The train is blocked from deploying to group2 (Wikipedia) due to a database performance regression. (T249964)
- Fri 2020-05-15 A solution for the performance regression has been found, however no major deploys on Friday.

Week of 18 May 2020

- Mon 2020-05-18 The patch for the performance regression is deployed. Impact to be confirmed next week when the train is deployed to group2/Wikipedia.
- There is no train schedule this week due to (virtual) team offsites.
- Production continues to be on 1.35.0-**wmf.31** for Wikipedia, and the newer 1.35.0-**wmf.32** for non-Wikipedia.

Fri 2020-05-22:

- 00:28 Configuration change 597895 "Enable coalesceKeys for non-global on commonswiki" is deployed.
- 00:28 **PROBLEM BEGINS**
- …
- 16:39 A thread in the Village Pump on Commons reports that some recently uploaded images can't be seen from Wikipedia.
- 18:08 Task T253405 was created from the VP thread.
- 18:16 Task T253408 reports a similar issue (duplicate).
- 19:00 CDanis looks into the issue but finds the earliest reported uploads seem to (now) work fine.

- 19:11 CDanis confirmed more recent reports and finds them to be reproducible. He also notices there is no correlation with the software version these wikis run (it seems to affect both Wiktionary and Wikinews on the newer version, and Wikipedia on last-week's version).
- …
- 20:02 RLazarus, CDanis and AaronSchulz are actively investigating.
- 20:10 Looking a probable causes in the upload system.
- 20:20 Looking a probable causes in the JobQueue, ChangeProp, chunked-uploading in UploadWizard.
- 20:29 Aaron finds the root cause in `ForeignDBViaLBRepo::getSharedCacheKey` which is forging local cache keys on behalf of Commons, from the execution context of another wiki.
- 21:19 Aaron uploads a code fix, https://gerrit.wikimedia.org/r/598118.
- 21:23 Krinkle has reviewed the fix while Aaron is re-creating the code fix for wmf.31 and for wmf.32. This was non-trivial because larger cross-cutting refactors landed both between wmf.31 and wmf.32, and in master since. And also because despite having no deployments in over a week, production was mid-train.
- 22:02 Aaron uploads code fixes to wmf.31 and to wmf.32.
- 22:24 Krinkle deployed the fixes. **PROBLEM ASSUMED SOLVED**.

Detection

The first known report (retroactively speaking) was in the Commons Village Pump. The first time we were aware of it was through a user report on Phabricator.

As far as we know, no alerts were fired at any time relating to this issue.

- Memcached was working fine from a service perspective. Usage levels and usage patterns were also normal.
- MediaWiki was operating fine.
- The frontends and health checks were all fine.

The one area were the issue may've been noticable to our monitoring is the HTTP traffic breakdown by status code.

When a user tries to acces a Commons file description page locally on a wiki, they would have gotten a 404 Not Found for the affected files. If this issue affected more than 10% of uploads, and if accessing such pages directly was commonly done by users for some reason, then it would have likely showed in the HTTP 40x respoonse monitoring. However, neither was the case.

Conclusions

What went well?

- The Memcached configuration change was applied to Beta Cluster first. But (see below).

What went poorly?

- It wasn't clear who, if anyone, is maintaining the UploadWizard extension, or core's FileRepo code, which is what makes Commons possible.
- There is no QA for uploading, thumbnails, multimedia embedding etc as far as we know. If we did, they would likely have noticed and reported it to us at least half a day earlier (either from Beta, or from a production/test wiki).
- Patches from today's codebase had to be ported to the master state of 2 weeks ago and 3 weeks ago, because production was left in a multi-version state over both a weekend and a full no-deploy week. Managing multple versions is inevitable to some extent, but in general Tue-Wed-Thu already feels like a long enough time to juggle two branches for, nevermind 2.5 weeks.
- The Memcached configuration change was applied to Beta Cluster first, but we did not do a staggered rollout there. The set of cirumstances required for this incident include the staggered rollout. In hindsight, we can't know if we would have noticed the issue in Beta. Out of all the possible features in production that use Memcached, it is unlikely the engineers would have specifically tested file uploads in Beta, or if they did that they would not have then tried to use the file on a Wikipedia page in Beta, and hit that 1:10 race condition.

Where did we get lucky?

- Aaron Schulz was around who 1) knows core's FileRepo code which is where the root cause was eventually found (an ancient bug), and 2) happens to also be the person who made the Memcached configuration change the day before. Aaron is likely the only person around who would have known that 1 faulty cache key in the middle of FileRepo's code as being related to the observable end-user problem, and who could then

confirm that the Memcached change the day before indeed had something to do with this.

How many people were involved in the remediation?

- 4 people. 2 SREs, and 2 software engineers. There was no incident commander.

Actionables

- Adopt a policy to by default prevent the train from pausing mid-way over a weekend. By Friday, either roll out or roll back. Weekend incident investigation should not have to deal with multi-version. Even if the train is not longer blocked and there simply wasn't time to roll out completely, either rollback or roll out anyway. (TODO: Create task)
- Determine stewardship for UploadWizard extension. (TODO: Create/find task) (mw:Developers/Maintainers lists mw:Readers/Structured Data; see also phab:T240281)
- Determine stewardship for MW Core's upload API. (TODO: Create/find task) (mw:Developers/Maintainers lists mw:Readers/Structured Data; see also phab:T240281)
- Determine stewardship for MW Core's FileRepo backend. (TODO: Create/find task) (mw:Developers/Maintainers lists mw:Readers/Structured Data; see also phab:T240281)
- Consider having some amount of regular QA for uploading and multimedia in Beta and prod. (TODO: Create task)

TODO: Add the #Wikimedia-Incident-Prevention Phabricator tag to these tasks.

Categories: Incident documentation | Incident documentation drafts