



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/20191210-updateCategoryCounts

[< Incident documentation](#)

document status: draft

Contents [\[hide\]](#)

- [1 Summary](#)
 - [1.1 Impact](#)
 - [1.2 Detection](#)
- [2 Timeline](#)
 - [2.1 What went well?](#)
 - [2.2 What went poorly?](#)
 - [2.3 Where did we get lucky?](#)
 - [2.4 How many people were involved in the remediation?](#)
- [3 Links to relevant documentation](#)
- [4 Actionables](#)

Summary

On 2019-12-10, mainly between 22:26 and 22:39 UTC (but there were other smaller instances in the previous hour), Common wiki database replicas were lagging behind, causing slowdown, returning stale results, errors and, as a consequence of the original bug, category counts were updated incorrectly.

Impact

The user-noticeable impact was directly felt on Commons for a few minutes due to articles being removed from `Category:11`. However, the bug was present production on the following wikis during the given dates:

- testwiki: 2019-11-26 19:06 to 2019-12-10 21:16
- rest of group 0: 2019-11-26 19:46 to 2019-12-10 21:23
- group 1: 2019-12-04 20:12 to 2019-12-04 23:38, 2019-12-09 17:08 to 2019-12-11 23:18
- group 2: 2019-12-09 17:52 to 2019-12-12 20:18

While the other wikis did not suffer from obvious impact (lag), they may have suffered from incorrect category counts.

Detection

First notice was on IRC (see timeline).

There is an alert for lag, but it only triggers after 300 seconds for 10 checks (while slowdown happen after lag is higher than 1 second, and hard down is after 6 seconds). Top production lag was around 100 seconds. [JCrespo thinks (but is unsure) that the reason for this discrepancy is because SRE concerns are only about broken mysql replication and excessive (e.g. 1 hours) lag, and no one attending application problems currently receives these alerts].

Timeline

- 22:12 - 22:14: Smaller spike of lag. Other less impacting occurrences may have happened before then with no user-noticeable impact.
- 22:26: For the purposes of this document, outage will be considered starting here, when a more continuous lag spike starts. This is due to the following query running:

[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](#)

```
UPDATE /* WikiPage::updateCategoryCounts */
`category` SET cat_pages = cat_pages - 1,
cat_subcats = cat_subcats - 1 WHERE cat_title = 11
```

This query is incorrect, the 11 should be enclosed between quotes (""). This causes a slow query (not using an index) plus all categories starting with 11 to be updated, rather than just 1.

- 22:36: on IRC:



```
2019-12-10 22:32:36 <Amir1> I'm getting db locked for Special:UploadWizard on
commons: https://commons.wikimedia.org/wiki/Special:UploadWizard
2019-12-10 22:32:48 <Amir1> non stop, for minutes now
2019-12-10 22:33:09 <Amir1> ok now
2019-12-10 22:33:30 <Reedy> https://commons.wikimedia.org/w/api.php?
format=xml&action=query&meta=siteinfo&siprop=dbrepllag&sishowalldb=1
2019-12-10 22:33:32 <Reedy> There seems to be lag
2019-12-10 22:34:38 <cdanis> there is a _lot_ of read traffic on s4 right now
2019-12-10 22:35:01 <wikibugs> (PS17) Jbond: puppet-merge: refactor [puppet] -
https:// Gerrit: https://gerrit.wikimedia.org/r/544214
2019-12-10 22:35:02 <cdanis> not sure where it is coming from, but, baseline
is something like 200k rps, but it's getting 6-7M rps right now
```

- 22:39 lag ceases. cdanis, Apergos and jcespo (maybe others) are responding to the issue. The long running query is detected through long running query monitoring and the above query is seen being executed, potentially many times. Long running query killer does not kill writes, as that can cause even a worse outage.
- Dec 11, 01:24: Ticket [Task T240405](#) is created.
- Dec 11, 23:19: UTC wmf.10 with a fix is rolled into commons
- Dec 13, 18:31: The code point is tested and the query is correct

What went well?

- for example: automated monitoring detected the incident, outage was root-caused quickly, etc

What went poorly?

- for example: documentation on the affected service was unhelpful, communication difficulties, etc

Where did we get lucky?

- The query was relatively harmless by not affecting canonical data (category counts). If it had affected main metadata or data, backups would have to be recovered in a non trivial operation

How many people were involved in the remediation?

- for example: 2 SREs and 1 software engineer troubleshooting the issue plus 1 incident commander

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

- Immediate issue to avoid recurrence of the same problem: [Task T240405](#)
- Followup #1: [Task T221795](#) to fix category counts
- Followup #1: [Task T108255](#) enable strict mode to prevent lose sql mode for MySQL
- Change some dangerous updates of a single row to be `LIMIT 1`?

Categories: [Incident documentation](#) | [Incident documentation drafts](#)

This page was last edited on 28 April 2020, at 19:54.

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

