



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/20180710-MediaWiki

[< Incident documentation](#)

Contents [\[hide\]](#)

- 1 [Summary](#)
- 2 [Timeline A](#)
 - 2.1 [Follow-up](#)
- 3 [Timeline B](#)
 - 3.1 [Root cause](#)
- 4 [Links to relevant documentation](#)
- 5 [Actionables](#)

Summary

Part of the rollout of mcrouter ([T192370](#), [T198239](#)), is to configure MediaWiki to write to both nutcracker (the old Memcached proxy) and mcrouter (the new proxy). The abstraction layer used for this (MultiWriteBagOStuff) had two distinct problems that each caused a partial outage at different times.

1. Around 2018-07-10 12:00 UTC; MultiWriteBagOStuff did not use the `makeKey()` logic from MemcachedBagOStuff, instead it used the default `BagOStuff::makeKey()`. This caused access to certain cache keys (specifically, those that contain non-ASCII characters) to consistently throw fatal exceptions. These caches are needed as part of the edit process, and as such, many pages became impossible to edit. **Impact:** For about 30 minutes, many edit attempts and logged-in page views were failing due to fatal exceptions resulting in a HTTP 500 response.
2. Around 2018-07-10 20:00 UTC; MultiWriteBagOStuff required that writes succeed on both backends before returning. This wasn't actually possible because both proxies communicate with the same group of memcached servers. Sending a `get()` or `set()` command via both proxies is fine, but letting both proxies perform a logical "add" operation would work if both proxies operate on the same backend, given only one can "win". This problem was hard to detect because it did not affect all "add" operations. The two proxies use a different HashRing (key-to-server mapping). It would fail only when when both proxies mapped a key to the same memcached backend. This mapping is deterministic and thus those keys would consistently fail for any MediaWiki server attempting to use it. **Impact:** For about 2 hours, snapshot servers were experiencing high rates of Memcached errors. Web traffic appeared unaffected.

Both issues had been seen before in testing and on Beta Cluster, and were already fixed in master and the group0 branch, but had not yet been back-ported to the group2 wmf branch.

Timeline A

See also <https://tools.wmflabs.org/sal/production?q=synchronized&d=2018-07-10>.

- 11:32 <aaron@deploy1001> Synchronized wmf-config/mc.php: Make all non-test wikis write to both nutcracker and mcrouter. <https:// Gerrit.wikimedia.org/r/440470>
- 11:36 [#wikimedia-operations] <Raymond_> Since a few minutes I am getting Fatal Error " [W0SZmgpAAD4AAFHXWLcAAACR] Exception" on Wikimedia Commons while saving.
- 11:43 <icinga-wm> PROBLEM - MediaWiki exceptions and fatals per minute on graphite1001 is CRITICAL: CRITICAL: 90.00% of data above the critical threshold [50.0]
<https://grafana.wikimedia.org/dashboard/db/mediawiki-graphite-alerts>

Analysis in Logstash showed that entries in the log for `type:mediawiki channel:exception` were almost exclusively due to " `Key contains invalid characters`". As explained in the summary, this is due to MultiWriteBagOStuff not using the correct `makeKey()` implementation, allowing non-ASCII characters to remain unescaped and thus causing the MemcachedBagOStuff class to throw a fatal error.

Before investigating further, the first action was to revert the mcrouter deployment.

[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](#)

- 12:06 <aaron@deploy1001> Synchronized wmf-config/mc.php: Revert "Make all non-test wikis write to both nutcracker and mcrouter".

At this point, general MediaWiki service recovered.

Follow-up

Logstash showed that:

- There were 7,476 entries in the " `type:mediawiki channel:exception` " log, between 11:00 and 13:00.
- Of those, 7,390 entries were for "Key contains invalid characters".
- Of those 3,745 (about half) were from SpamBlacklist.

While non-ASCII characters are supported, they should be avoided if their origin is user input. This was fixed in SpamBlacklist by <https://gerrit.wikimedia.org/r/444856> and deployed subsequently to help avoid future regressions.

- 12:20 <aaron@deploy1001> Synchronized php-1.32.0-wmf.12/extensions/SpamBlacklist/includes/SpamBlacklist.php: 583dc7a92f9b
- 12:24 <aaron@deploy1001> Synchronized php-1.32.0-wmf.10/extensions/SpamBlacklist/includes/SpamBlacklist.php: 08a2153f7aa

The general issue of MultiWriteBagOStuff not using the correct `makeKey()` implementation was fixed shortly after by <https://gerrit.wikimedia.org/r/#/c/mediawiki/core/+442245/>, and also deployed.

- 12:53 <aaron@deploy1001> Synchronized php-1.32.0-wmf.10/includes/libs/objectcache/ReplicatedBagOStuff.php

With this root cause found, fixed and deployed; the mcrouter deployment was unblocked and would be re-tried later that day (see below).

Timeline B

See also <https://tools.wmflabs.org/sal/production?q=synchronized&d=2018-07-10>.

- 18:33 <aaron@deploy1001> Synchronized wmf-config/mc.php: Make all non-test wikis write to both nutcracker and mcrouter (again).

For the first hour following this deployment, MediaWiki service appeared to be fine, and there were no icinga alerts.

Looking directly at MediaWiki exceptions in Logstash ([link](#)) also showed no new exceptions, nor any increase in rate.

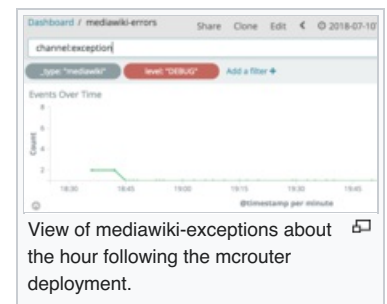
- 19:16 <icinga-wm> PROBLEM - High CPU load on API appserver on mw1227 is CRITICAL

About 45 minutes after the deployment, Icinga reported an issue with one API app server. This didn't stand out in #wikimedia-operations given a fair amount of time had passed and people were now working on other things. 30 minutes later, Icinga reported dozens of more API servers having issues, as well end-user traffic from Varnish being affected.

- 19:50 <icinga-wm> PROBLEM - High CPU load on API appserver on mw1226 is CRITICAL
- 19:55 <icinga-wm> PROBLEM - High CPU load on API appserver on mw1222 is CRITICAL
- 19:55 <icinga-wm> PROBLEM - Text HTTP 5xx reqs/min on graphite1001 is CRITICAL: CRITICAL: 22.22% of data above the critical threshold [1000.0] https://grafana.wikimedia.org/dashboard/file/varnish-aggregate-client-status-codes.json?panelId=3&fullscreen&orgId=1&var-site=All&var-cache_type=text&var-status_type=5
- 19:56 <icinga-wm> PROBLEM - HTTP availability for Varnish at eqiad on einsteinium is CRITICAL: job=varnish-text site=eqiad <https://grafana.wikimedia.org/dashboard/db/frontend-traffic?panelId=3&fullscreen&refresh=1m&orgId=1>
- etc.

At this point, Aaron and Timo started investigating.

The "Production Logging" dashboard [showed](#) a big increase in MediaWiki errors (not exceptions). From under 10 per minute normally, to 200 per minute. The error rate had been elevated since an hour earlier at 18:35 UTC (minutes after the deployment). Looking in Logstash for mediawiki-errors ([permalink](#)) revealed that the majority of these errors were tagged `channel:memcached`. Their rate was normally under 5 per minute, but increased



to 200 per minute at 18:35, and increased again to ~800/min when the Icinga alerts started at 19:50.

The 10-minute burst of errors that started at 19:50, with 600 additional errors per minute, came from servers responding to "/w/api.php" (presumably API app servers). Mostly in the form "Memcached error for key "{memcached-key}" on server "{memcached-server}": SERVER ERROR ". No one key or key group stood out, besides what appear to be generally more popular keys (not more likely to fail). Including "wikibase_shared/1_32_0-wmf_10-wikidatawiki-hhvm:CacheAwarePropertyInfoStore " (25%) various "WANCACHE:v:<key>:lock" keys.

The remaining 200/min were from snapshot1008 (this is where command-line scripts run to generate Dumps). All in the form "Memcached error for key "{memcached-key}" on server "{memcached-server}": SERVER HAS FAILED AND IS DISABLED UNTIL TIMED RETRY " for the same key: "WANCACHE:v:global:lag-times:1:db1071:1:lock".

Action: There should have been an alert for this increased error rate. That would've revealed the problem an hour earlier.

The "Edit Count" dashboard showed that edits were affected for about 10 minutes. Edit rate was stable around 1250/min up until 19:50 UTC. It then dropped to a low of 500/min at 19:56, and went up again until it reached normal levels at 20:00.

- 20:00 <icinga-wm> RECOVERY - High CPU load on API appserver on mw1225 is OK
- 20:03 <icinga-wm> RECOVERY - HTTP availability for Varnish at eqiad on einsteinium is OK: All metrics within thresholds. <https://grafana.wikimedia.org/dashboard/db/frontend-traffic?panelId=3&fullscreen&refresh=1m&orgId=1>

Question: The app server and Varnish alerts recovered for unknown reasons. The MediaWiki error rates also went from 800/min back to 200/min for unknown reasons.

Question: It is unclear why Edit Count was affected. It is also unclear why it recovered within 10 minutes as no action was taken later.

Root cause

After about 30 minutes into the investigation we realised that this may be related to an issue we found on Beta Cluster last week, which had been fixed in master and part of the 1.32.0-wmf.12 train deploy for this week (which had just gone to group0), but was not yet backported to 1.32.0-wmf.10.

The issue in question (T198280) was that the Memcached "add()" operations, commonly used for locks, would consistently fail (when using MultiWriteBagOStuff) for a seemingly random subset of cache keys (specifically, keys for which the hash ring of both Nutcracker and Mcrouter map to the same backend server).

This was made much worse by the fact that MultiWriteBagOStuff::merge() always used mergeViaLock (instead of mergeViaCas). Memcached add() is relatively rarely used by MediaWiki, but Memcached merge() is incredibly common because it is what WANObjectCache::set() uses internally for its lock keys. This was fixed seen in Beta Cluster the week before, and already fixed in master with <https://gerrit.wikimedia.org/r/442813>. Backporting this patch resolved the issue.

- 20:48 <aaron@deploy1001> Synchronized php-1.32.0-wmf.10/includes/libs/objectcache/MultiWriteBagOStuff.php: 4fba9f6a032
- 20:50 <aaron@deploy1001> Synchronized php-1.32.0-wmf.10/tests/phpunit/includes/libs/objectcache/MultiWriteBagOStuffTest.php: 4fba9f6a032

Deploying this immediately resulted in the MediaWiki error log rates going back from 200/min to under 10/min.

Links to relevant documentation

Recent deployments:

- https://wikitech.wikimedia.org/wiki/Server_Admin_Log

Monitoring and profiling:

- <https://logstash.wikimedia.org/app/kibana#/dashboard/mediawiki-errors>
- <https://logstash.wikimedia.org/app/kibana#/dashboard/memcached>
- <https://grafana.wikimedia.org/dashboard/db/prometheus-varnish-http-errors>
- <https://grafana.wikimedia.org/dashboard/db/production-logging>
- <https://grafana.wikimedia.org/dashboard/db/apache-hhvm>
- <https://performance.wikimedia.org/php-profiling.html>

Actionables

- [phab:T199479](#) - Add alerts for Logstash rates in production.

Category: [Incident documentation](#)

This page was last edited on 9 November 2018, at 01:00.

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

[Privacy policy](#) [About](#) [Disclaimers](#) [Code of Conduct](#) [Developers](#) [Statistics](#) [Cookie statement](#) [Mobile view](#)
[Wikitech](#)

