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/20170320-Special AllPages

< Incident documentation

Summary

On the 2017-03-20, Special:AllPages started creating large amounts of slow database queries on enwiki, creating slowdowns/outage on all database queries on that wiki between 14:21 and 14:36 UTC. To mitigate this, AllPages was first fully disabled on all wikis, and later re-enabled with reduced functionality.

Timeline

- [not noticed at the time] Between 12-14 hours, scraping of [[Special:AllPages]] is done on some wikis- while this is a discouraged practice (both dumps, API or labs replicas are preferred options), it doesn't create any issues- partly because probably the throughput was probably low, partially because this is not a taxing operation for non-large wikis. This can be seen as a small spike of x4 s7 open connections at 12:23, maybe others. Also a user said that had happened.

- At 14:01, the same scraping starts impacting enwiki, increasing the number of open connections (not yet in outage, just making things slow)

- At 14:21, a first page is sent about db1080 unable to receive checks because max_connections (10000) is full. All other databases serving main traffic follow, with a period of flopping alters as a) unreachable databases are depooled, which makes the connections drop, and then repooled, b) a watchdog kills long running connections, but only when they reach 300 seconds (and the concurrency on queries is high enough to not being able to keep with the new connections). A snowball effects happens because the more connections are created, the slower the servers are, and the less they can keep processing or killing queries.

- At this point, this is thought to be a software issue for the following reasons: it is happening on all database servers at the same time, the aggregated open connections is high with no visible db functionality problem or locking other than that, and the slow query logging only show a single query digest being slow (shown also on SHOW PROCESSLIST): `SELECT /* SpecialAllPages::showChunk */ page_title FROM `page` WHERE page_namespace = '0' AND page_is_redirect = '0' AND (page_title < 'Entomacrodus_stellifer') ORDER BY page_title DESC LIMIT 344, 1 /* bff89ebd2be06faa4ae5c18dcc972d02 db1079 eswiki 5s */`. This query is happening not only on enwiki, but it is only causing an outage there.

- At 14:22: hashar@tin: Synchronized php-1.29.0-wmf.16/includes/specials/SpecialWatchlist.php: reverts commit SpecialWatchlist.php 0d675d2 (duration: 00m 43s). This revert was requested, as it was a change that had happened just before involving a special page- the real Special page responsible has not yet

Evolution of enwiki open connections on the 2017-03-20

Evolution of 5xx http errors on the 2017-03-20

Evolution of Mediawiki exceptions on the 2017-03-20

Rate of long running queries from SpecialAllPages-showChunk on the 2017-03-20

identified. It has no effect.

- Conversation switches to private channels for the fear this could be a DOS attack. Special:AllPages is identified as the immediate cause by both inspecting the code and analyzing http requests done.
- At 14:36 hashar: Disabled Special:AllPages on all wikis making it spurts a blank page instead. (https://gerrit.wikimedia.org/r/#/c/343647/ 🖉)
- At 14:39, open connections return to normal levels. Outage and/or slowdown ends.
- T160914 is created as a private task because of fear of a directed attack- also to prevent malicious agents to simulate the same requests and bring the service down again. Discussion is ongoing about the root cause and the best way to proceed (the goal is to reenable Special:AllPages ASAP, once the issue has been avoided. A less detailed, but more informative public bug is created on T160916
- At 20:12 some proper patches (https://gerrit.wikimedia.org/r/343681 🖉) are reviewed and deployed to production, which reenables Special:AllPages but limits the ability to filter by is_redirect flag for wikis running in miser mode.

Conclusions

A long running database query that was usually infrequently used started being executed hundreds of times per minute, with >4000 executions. This lead to an exhaustion of db connection resources, and db resources in general on the shard "s1". While no server crashed, this meant a slowdown or denial of service of some mediawiki requests to the English Wikipedia. The exact amount of users affected is difficult to measure: around 600 errors were registered, but that doesn't account for requests that were served but were too slow to be useful, or requests that did not even arrive or were sent in the first place because of slowness. On the other side, some of those errors only affected the single user creating those requests. In any case, this mostly impacted editors and authenticated users' requests, as those are the mostly affected by uncached requests. We believe not most of the editors were affected, as the edit rate didn't seem highly affected at 14:20 hours 🖉 and contributors commented the site may be slow 🖉, but not unresponsive, and they could not reproduce a full outage reliably.

The first kind of actionables is to fix the query so it is not slow anymore or it is avoided in the first place.

The second kind are longer-term tasks to avoid this kind of issues in the first place, by implementing (or at least discussing the viability of) several strategies, at several layers: varnish cache, mediawiki application, and databases.

Actionables

Direct actionables to prevent this issue from reappearing

- Solve ongoing connection database issues Task T160914 ✓ **Done**
- Reenable Special:AllPages, once the core issue is solved or worked around it Task T160916 ✓ **Done**
- Optimize SpecialAllPages::showChunk so that filtering based on the redirect flag may work again Task T160983 ⟳ **In progress**

Long term actionables to solve more generic issues (to be discussed)

- Throttling access to Special Pages that make potentially expensive queries Task T160920 ⟳ **In progress**
- Reduce max execution time of interactive queries or a better detection and killing of bad query patterns. Task T160984 ⟳ **In progress**
- Create an easy to deploy kill switch for every self-contained mediawiki functionality Task T160985 ⟳ **In progress**
- BBlack mentioned some pending work on improve caching (Ticket?)

Category: Incident documentation