Page   Discussion

Read   View source   View history

Search Wikitech

Incident documentation/20150403-API

< Incident documentation

Contents [hide]

Summary

The Wikimedia API was degraded in performance and ultimately became unresponsive from approx. 07:48 UTC to 08:08 UTC. api.php URLs were slow to load and/or result in HTTP 503 errors.

This is believed to have been caused by a cascading failure due to the following chain of events:

- An svwikivoyage Modul:Wikibase edit ⧉ happens with a Lua syntax error (infinite recursion on is_defined); quickly reverted five minutes later ⧉.
- Reparsing jobs get inserted into the job queue and subsequently executed.
- MediaWiki jobrunners execute + the Parsoid cluster request api.php.
- Scribunto generates parsing errors and emits them, as evident by:

  - ```
    2015-04-03 08:38:28 mw1203 svwikivoyage: MessageCache::parse called by
    Scribunto_LuaError::getScriptTraceHtml/Message::parse/Message::toString/Message::parseText/MessageCache::parse
    with no title set.
    ```

- `MessageCache::parse` cause the following queries, both returning an empty resultset:

  - ```
    SELECT /* DatabaseBase::selectRow 10.64.32.84 */  ips_item_id  FROM `wb_items_per_site`   WHERE ips_site_id
    = 'svwikivoyage' AND ips_site_page = 'API'  LIMIT 1
    ```
  - ```
    SELECT /* DatabaseBase::selectRow 10.64.32.84 */  ips_item_id  FROM `wb_items_per_site`   WHERE ips_site_id
    = 'svwikivoyage' AND ips_site_page = 'Dwimmerlaik'  LIMIT 1
    ```

- The above queries storm s5 slaves with hundreds of thousands of QPS (tendril showed db1071 alone with ~270.000 QPS, although that's probably an overestimation)
- s5 slaves get overloaded; API appservers start emitting 500s.

Moreover, it was observed that jobrunners & API appservers continued to emit those warnings and queries above even long after the outage was over. All of them continued to query a depooled slave (db1049) and a particular server continued to query that particular slave continuously even after it was depooled of traffic from Pybal.

Timeline

- [07:41] svwikivoyage's Modul:Wikibase gets changed with bad edit ⧉ (infinite recursion)
- [07:48] Icinga emits the first individual appserver (non-paging) error
- [07:49] Icinga pages the outage
- [07:49] Max starts investigating and tries to figure out how to text someone in Europe with his crappy US plan
- [07:50] Watchmouse pages the outage
- [07:52] Faidon begins to investigate
- [08:01] Database overload and offending queries found
- [08:03-4] Giuseppe & Mark join the investigation
- [08:05] Icinga individual appserver recoveries start flowing
- [08:05] Watchmouse pages the recovery
- [08:08] Icinga pages the recovery; outage ends
- [08:08-09:10] Investigation into the root cause continues
- [09:32] Sean depools db1049 (an s5 slave)
- [10:20] Faidon restarts HHVM in all API & jobrunners after reports by Sean of continued db1049 activity with the queries above; activity stops immediately.

Conclusions

- There seems to be a LuaSandbox bug that caused a multiplication factor into this bug and warrants further investigation.
- Significant time was wasted during the investigation grepping access logs, as the initial assumption was that "Dwimmerlaik" was a page name and hence user-triggered. However, "Dwimmerlaik" is a placeholder identifier in MediaWiki's code (includes/cache/MessageCache.php).
- Both "API" and "Dwimmerlaik" seem to be placeholders for "ghost titles" (as the comment in the code suggests). Those two placeholders, however, cause database queries that are probably redundant and possibly harmful.
- A user-triggered Lua error should not cause a globaltitlefail.log warning and two redundant queries.

Actionables

- Merge https://gerrit.wikimedia.org/r/#/c/201678 ⧉
- Set up alerts on API response times and DB query rates. There were significant response time increases a few hours earlier (99th percentile for simple revision requests ~500ms --> 9s for two ~45 minute blocks), which in hindsight should probably have been investigated more thoroughly. We saw a reduction in load on the API cluster and Parsoid around the time, but didn't have the information about database query rates readily available to identify the problem.

Category: Incident documentation