Page   **Discussion**

Read   **View source**   **View history**

Search Wikitech

Incident documentation/20180226-WikibaseQualityConstraints

< Incident documentation

Contents [hide]

Summary

When caching of constraint check results was enabled on Wikidata, some API requests would trigger SQL queries on the `page` , `revision` and `text` tables without any `WHERE` clause. This resulted in extremely high server load of general-purpose replica servers in s8.

Timeline

All times in UTC.

- 14:22 – `zeljkof` deploys `413724` to mwdebug1002 as part of SWAT.
- 14:22-14:26 – Lucas Werkmeister (`Lucas_WMDE`) tests the change on Wikidata. It seems to work as expected – successive calls of `wbcheckconstraints` on the same entity are much faster, since the result is read from cache.
- 14:27 – Lucas Werkmeister (`Lucas_WMDE`) reports back to `zeljkof` and confirms the change is ready to deploy.
- 14:28 – `zeljkof` deploys the change. Server admin log entries during this time
- 14:28 – eth0 traffic on db1109 immediately begins to grow, crossing 600 Mbps within the next ten minutes (the normal level is around 50 Mbps) – see Grafana . db1092 and db1104 are also affected. CPU load, number of running processes, and disk I/O also increase accordingly. However, according to `jynus`, "the slowdown was slow to buildup, so it was not detected by monitoring immediately".
- 17:24 – `marostegui` and `jynus` comment on phabricator:T184812, pointing to the massive spike in Wikidata replicas (Grafana).
- 17:34 – `jynus` "deploying new query killer to db1109". Server admin log entries during this time
- 17:41 – Chad deploys a revert of the config change (`414720`). log entry (stashbot failed to log this and some other messages around that time to wiki)
- 17:41 – eth0 traffic, CPU load etc. begin to drop again (Grafana).
- 17:41 – `jynus` "I am going to kill [queries?] [a]gain to see they don't come back"
- 17:44 – eth0 traffic, CPU load etc. back to normal levels (Grafana).
- 17:45 – `jynus` "things seem under control"

Conclusions

Due to a logic bug (phabricator:T188384), WikibaseQualityConstraints' `CachingResultsBuilder` asked Wikibase' `WikiPageEntityMetaDataLookup` for the latest revision information of an empty list of entities. The `WikiPageEntityMetaDataLookup` had a special safeguard (added in `294340`) for this case to avoid costly queries, adding a condition `"0"` . However, `Database::selectSQLText` actually turns this condition into a query with no `WHERE` clause (instead of something like `WHERE FALSE` , which `WikiPageEntityMetaDataLookup` probably intended). WikibaseQualityConstraints should pay more attention to special cases when requesting entity IDs (empty list, long list); Wikibase should make sure that safeguards actually work as intended; and core should not let simple programming errors result in completely unlimited queries.

It is also problematic that the incident apparently went undetected for almost three hours, even though it was visible in Grafana within minutes of deployment, but I don't know why this was possible, or what should have prevented it.

Actionables

- WikibaseQualityConstraints: only filter for result statuses after collecting metadata. [phab:T188384](#)
- WikibaseQualityConstraints: don't pass an empty list of entity IDs to the `WikiPageEntityMetaDataAccessor`. [phab:T188311](#)
- WikibaseQualityConstraints: don't pass overly *long* lists of entity IDs there, either. [phab:T188312](#)
- WikibaseQualityConstraints + Wikibase: don't join `page`, `revision` and `text` when we just need `page.page_latest`. [phab:T188381](#)
- Wikibase: fix the safeguard for an empty list of conditions in `WikiPageEntityMetaDataLookup`. [phab:T188313](#)
- core: only `$conds = []` should result in a missing `WHERE` clause, not any other value that is `empty` according to the PHP [empty](#) function. [phab:T188314](#)
- monitoring: detect high server load earlier – `jynus` "a prometheus alert would be nice" [phab:T188317](#)

Category: Incident documentation