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

Incident documentation/20181001-DeleteLocalPasswords

< Incident documentation

Contents [hide]

Summary

The MediaWiki maintenance script (T201009 / T201009, DeleteLocalPasswords.php) did not include a `waitForReplication` call. When run on certain databases (commons and dewiki), it updated over a million rows, causing an extended period read only time and slowdown of requests on Commons (and other projects using Commons database resources) approximately between 10:48 and 11:01 and on the German Wikipedia between 11:02 and 11:16 UTC.

Timeline

SAL for 2018-10-01: Script starts (this wasn't known at the time of the initial response)

```
10:10 <Amir1> mwscript
extensions/CentralAuth/maintenance/deleteLocalPasswords.php --wiki=fawiki --
delete (T201009)
10:15 <Amir1> ladsgroup@mwmaint2001:~$ mwscript
extensions/CentralAuth/maintenance/deleteLocalPasswords.php --prefix on all
CentralAuth wikis (T201009)
```

First alert on IRC:

```
10:51:16 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s4 on db2058 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 483.80 seconds
```

(thinking it is a server-only issue, and seeing no slow query ongoing but 'slow' loadgroup queries db2058 is depooled and the 2 slow queries killed)

```
10:58:57 <logmsgbot>  !log jynus@deploy1001 Synchronized wmf-config/db-
codfw.php: Depool db2058 (duration: 00m 57s)
```

But other hosts starts alerting, too:

```
11:04:46 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s4 on db2090 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 310.07 seconds
11:05:37 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s4 on db2073 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 301.54 seconds
11:05:56 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s4 on db2084 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 318.34 seconds
```

Swat is paused, probably a master or code bug issue is thought. Main queries happening at the moment from replication are DeleteLocalPassword ones from mwmaint2001. Discussion happens if to kill the commons script, and it is done (but the issue just jumps into dewiki).

```
11:06:57 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s4 on db2073 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
11:07:16 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s4 on db2084 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
11:08:12 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s4 on db1097 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
11:08:36 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s4 on db2090 is OK: OK
slave_sql_lag Replication lag: 0.05 seconds
11:09:17 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s4 on db2058 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
```

Finally, the parent process is found:

```
11:09 <_joe_> killed bash runner.sh by user ladsgroup on mwmaint2001
```

dewiki ongoing process is killed at the same time, causing temporary lag, too (but not enough to page):

```
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1102 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 440.58 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1110 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 441.13 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1124 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 445.59 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1082 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 452.27 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1097 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 464.86 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1100 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 466.61 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1096 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 474.90 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1070 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 334.97 seconds
11:17:01 <icinga-wm>  PROBLEM - MariaDB Slave Lag: s5 on db1113 is CRITICAL:
CRITICAL slave_sql_lag Replication lag: 490.54 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1110 is OK: OK
slave_sql_lag Replication lag: 59.86 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1124 is OK: OK
slave_sql_lag Replication lag: 0.43 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1082 is OK: OK
slave_sql_lag Replication lag: 0.03 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1097 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1100 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1096 is OK: OK
slave_sql_lag Replication lag: 0.16 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1070 is OK: OK
slave_sql_lag Replication lag: 0.10 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1113 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
11:17:01 <icinga-wm>  RECOVERY - MariaDB Slave Lag: s5 on db1102 is OK: OK
slave_sql_lag Replication lag: 0.00 seconds
```

Conclusions

- Do not forget to add `waitForReplication` calls.
- Registering the maintenance on the Deployments page might have helped to identify the issue more quickly.
- The script ran successfully on fawiki (medium-sized). Is there some way to notice that a script is causing replag (assuming it did so at all) when it is not large enough to bring things down?

Links to relevant documentation

Not sure what would be relevant here. waitForReplication()?

Actionables

- Consider adding code to the MediaWiki DB layer to detect huge numbers of writes without waiting for slaves - phab:T205893
- Consider adding dry-run options on all maintenance scripts, even those that seem trivial in the future
- Communicate more clearly to sysadmins (people that can respond quickly to incidents) of what, where and when maintenance scripts are done as documented at https://wikitech.wikimedia.org/wiki/Deployments (Long running changes/scripts) so the script can be easily and faster killed if an unexpected issue arises

Category: Incident documentation