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/20181016-Replication gap on s8 eqiad hosts

< Incident documentation

Summary

While codfw was the active DC (and writes were happening there), eqiad s8 (wikidatawiki) master (db1071) had a replication gap from 2018-09-13 09:08:17 to 2018-09-13 09:58:26 where all the data inserted for all the tables on codfw never reached eqiad.

Timeline

Times un UTC

- 12th Sept 14:52: Codfw becomes the active DC
- 13th Sept 08:01: Replication manually eqiad -> codfw disconnected (this is a normal procedure to avoid maintenance on the passive DC affecting the primary one)
- 13th Sept 08:36: Deployed schema change on s6 master - T89737
- 13th Sept 08:49: Deployed schema change on s5 eqiad master (db1070) - T89737
- 13th Sept 09:08: Deployed schema change on s8 eqiad master (db1071) - T89737
- 13th Sept 09:08:17: Events stop to get replicated from codfw master to eqiad master (db1071)
- 13th Sept 09:09: db1071 starts lagging (this is normal as replication is manually stopped for the schema change)
- 13th Sept 09:44: Enable GTID on eqiad masters (unfortunately we don't know the exact timing when s8 was done)
- 13th Sept 09:58:26: Events start to get replicated again from codfw master to eqiad master (db1071)
- 13th Sept 10:27:54 GTID gets enabled on db1071 (the command to enable GTID: *STOP SLAVE; CHANGE MASTER TO MASTER_USE_GTID=Slave_pos; START SLAVE;* And I/O threads positions to a different position (it should have used 216202297 and it used 1036765620)

```
180913 10:27:53 Slave SQL thread exiting, replication stopped in log 'db2045-
bin.005879' at position 1036765620
180913 10:27:53 [Note] Slave SQL thread exiting, replication stopped in log
'db2045-bin.005879' at position 1036765620
180913 10:27:53 [Note] Slave I/O thread exiting, read up to log 'db2045-
bin.005880', position 216202297
180913 10:27:54 [Note] 'CHANGE MASTER TO executed'. Previous state
master_host='db2045.codfw.wmnet', master_port='3306', master_log_file='db2045-
bin.005880', master_log_pos='216202297'. New state
master_host='db2045.codfw.wmnet', master_port='3306', master_log_file='db2045-
bin.005879', master_log_pos='1036765620'.
180913 10:27:54 [Note] Previous Using_Gtid=No. New Using_Gtid=Slave_Pos
180913 10:27:54 [Note] Slave I/O thread: Start semi-sync replication to master
'repl@db2045.codfw.wmnet:3306' in log 'db2045-bin.005879' at position 1036765620
```

- 13th Sept 11:13: We believe the schema change finished

- 10th Oct 14:18: Eqiad becomes the active DC
- 11th Oct 08:56: Addshore pings the DBA about possible data drift on IRC and on phab - https://phabricator.wikimedia.org/T206743#4657812
- 11th Oct 09:00: DBAs confirm there are rows not present in eqiad and investigation is started
- 11th Oct 11:13: DBAs put a plan of action together to address the issue: https://phabricator.wikimedia.org/T206743#4658146 by: filling in rows from codfw (even though data might be inconsistent) and by recloning hosts from codfw as the data there is consistent.
- 11th Oct 14:19: All eqiad hosts get the rows refilled. At this point there is no missing data.
- 11th Oct 14:57: First recentchanges host db1099 is recloned and pooled in production
- 11th Oct 18:16: Second recentchanges host db1101 is recloned and pooled in production

DBAs decided to clone the remaining hosts in a faster way on Monday when there is more coverage.

- 12th Oct 18:25: Addshore runs a script to get some page fixed and get more consistency https://phabricator.wikimedia.org/T206743#4662028
- 15th Oct 05:00 - 13:28: All the replicas in eqiad (apart from db1087, which is depooled) are recloned and in production. From this point data is consistent and safe.
- 15th Oct 14:00: DBAs start to discuss how to get db1087 (and labs fixed) in a safe and fast way.
- 16th Oct 14:00: Fixing db1087 (labs master) starts. All tables apart from pagelinks and wb_terms get checked and fixed.
  - pagelinks and wb_terms require more time because they are so big that they cannot be diffed all at once as the mysql clients runs out of memory, so the table needs to be split.
- 17th Oct 06:52: Starting to fix the master (db1071)
- 17th Oct 12:10: s8 master (db1071) fixed (only pending the big tables pagelinks and wb_terms which will take more time)
- 17th Oct 19:04: s8 master gets pagelinks fixed, wb_terms is half way done.
- 18th Oct 16:05: s8 core hosts finished getting fixed (pending labs)
- 19th Oct 10:00: pagelinks table started to get re-imported on labs
- 22th Oct 11:38: Labs wb_terms is fixed, at this point everything should be consistent. A new round of checks is started to double check (T206743#4685983)
- 23th Oct 05:49: Second round of tables started to get checked: T206743#4687946
- 23th Oct 08:37: abuse_filter_log and change_tag that reported differences get fixed
- 24th Oct 08:13: All tables checked and confirmed consistent (db1092, host cloned from codfw vs db1087/db1071 hosts fixed manually)

Conclusions

- For some reason replication jumped ahead on db1071 (s8 master) and skipped 50 minutes of transactions
  - We don't know why it happened. So far the theory points to: slave lag + gtid enablement + out of band schema change
  - We do pretty much every day out of band changes, but GTID is always enabled, so doing it with GTID disabled and then get it enabled isn't something we do that often, so it could be a corner case.
- Replication never broke - which is strange, specially on those slaves (db1124) where we use ROW based replication, which would have broken if a missing row was UPDATED or DELETE, which never happened (coincidence?)
- We didn't notice the issue until this was reported by an user.
- Only s8 was affected, the rest of sections were checked and confirmed clean (even though the same alter was applied to s5 and s6 eqiad masters)

Links to relevant documentation

Not much yet: Main tracking task: https://phabricator.wikimedia.org/T206743

ALTER tables run:

```
stop slave;
SET SESSION innodb_lock_wait_timeout=1; SET SESSION lock_wait_timeout=30;
set global innodb_online_alter_log_max_size = 3334217728000;
set session sql_log_bin=0;
ALTER TABLE /*_*/change_tag MODIFY ct_log_id int unsigned NULL, MODIFY ct_rev_id
int unsigned NULL;
ALTER TABLE /*_*/page_restrictions MODIFY pr_user int unsigned NULL;
ALTER TABLE /*_*/tag_summary MODIFY ts_log_id int unsigned NULL, MODIFY
ts_rev_id int unsigned NULL;
ALTER TABLE /*_*/user_newtalk MODIFY user_id int unsigned NOT NULL default 0;
ALTER TABLE /*_*/user_properties MODIFY up_user int unsigned NOT NULL;
ALTER TABLE /*_*/bot_passwords MODIFY bp_user int unsigned NOT NULL;
start slave;
```

Actionables

Still to be discussed:

- Was this a bug in Slave_Pos+schema change out of band + enabling GTID? Wouldn't be surprising as per: https://jira.mariadb.org/browse/MDEV-12012
- Some automatic table checksum comparison not only between replicas but between DCs for very concurrent tables would have alerted on data drifts: phab:T207253

Category: Incident documentation