

Quassel IRC - Bug #1512

SQLite journal/concurrency issues: SqliteStorage::getBufferInfo(): received more then one Buffer!

01/30/2019 01:12 PM - mgorny

Status:	New	Start date:	01/30/2019
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Quassel Core	Estimated time:	0.00 hour
Target version:		OS:	Any
Version:	0.13.0		
Description			
Every few days, my core crashes with an error like:			
<pre>2019-01-30 11:19:09 [Error] SqliteStorage::getBufferInfo(): received more then one Buffer! 2019-01-30 11:19:09 [Error] Query: "SELECT bufferid, buffertype, groupid\nFROM buffer\nWHERE buffer.networkid = :networkid AND buffer.user id = :userid AND buffer.buffercname = :buffercname" 2019-01-30 11:19:09 [Error] bound Values: 2019-01-30 11:19:09 [Error] 0 : cynede_afk 2019-01-30 11:19:09 [Error] 1 : 1 2019-01-30 11:19:09 [Error] 2 : 1 2019-01-30 11:19:09 [FATAL] ASSERT: "false" in file /home/mgorny/git/quassel/src/core/sqlitestorag e.cpp, line 1176</pre>			
I have no clue why this would happen. Is it possible that the database was corrupted as a result of earlier crashes?			
Can we have some nicer behavior here? Is there a nice way to fix the database by hand?			
Finally, can we enforce this via database constraints?			

History

#1 - 02/07/2019 12:13 AM - digitalcircuit

- Subject changed from [Error] SqliteStorage::getBufferInfo(): received more then one Buffer! to SQLite journal/concurrency issues: SqliteStorage::getBufferInfo(): received more then one Buffer!

- Category set to Quassel Core

Details

I agree that this should be enforced by database constraints, that Quassel should handle this more gracefully, and I'm not sure how Quassel would get into this state.

My *guess* is Quassel crashed in the middle of a write before it completed, or Quassel's current locking mechanism in SQLite allowed concurrent writes. Based on the bound values, it seems like a private message buffer was renamed (either due to /nick changes, or manual action). To fix this and other issues, Quassel likely needs to adopt [SQLite's WAL mode](#), which had performance issues with older SQLite versions (e.g. as in Ubuntu 14.04). With 0.14/master increasing build requirements, this is something we can finally try to tackle.

Crashes once this inconsistency exists are likely merely periodic as it'd only occur when Quassel tries to get this buffer information, which may be when a particular user logs in, etc.

Workaround/recovery

To recover, you should be able to find the duplicate buffer and rename it:

1. Find the IDs of the duplicate buffers, the ones with the same name/user ID/network ID

With the original query of...

```
SELECT bufferid, buffertype, groupid FROM buffer WHERE buffer.networkid = :networkid AND
buffer.userid = :userid AND buffer.buffercname = :buffercname;
```

...and the bound values of...

cynede_afk, 1, 1

This should be...

```
SELECT bufferid FROM buffer WHERE buffer.networkid = 1 AND buffer.userid = 1 AND buffer.buffercname = 'cynede_afk';
```

2. Rename one of the buffers to avoid the collision

Make a backup first! I'm not 100% certain this will fix the issue.

Rename the buffercname and buffername of the problematic buffer to a unique name...

```
UPDATE buffer SET buffercname = 'cynede_afk_dup', buffername = 'cynede_afk_dup' WHERE bufferid = XXXX;  
-- Replace XXXX with one of the buffer ID numbers from above
```

Retry the query in step 1. If there's still more results, repeat this, but pick different names for buffercname and buffername, e.g. cynede_afk_dup2.

#2 - 02/07/2019 08:12 AM - mgorny

Thank you. I've just applied the recovery and we'll see what happens now. It turned out that `cynede_afk` had two duplicates with very different numbers (one 1xx, other 4xx), so I suspect this might have been going for some time already. Though I suspect I never knew because I haven't been running debug builds before.

#3 - 02/14/2019 05:48 AM - digitalcircuit

Thanks for feedback; I'm glad the recovery worked! That does seem odd, and given the assert() only on debug builds, it's a relief it's likely not a new issue/regression.

Quassel should still handle this better, as originally agreed.

Pardon my delayed reply; turns out I need to manually watch an issue after commenting in order to get email notifications.

#4 - 03/31/2019 07:15 PM - mgorny

It happened to me today, and it seems that there is some deeper issue that is causing those duplicate buffers to occur.

More specifically, I received messages from '<person>', and after applying the deduplication as explained above, the <person>_dup buffer contained only the messages received today. If that's of any help, the messages were described as '<person>1', so I think the person in question may have changed the nickname just after writing.

#5 - 10/03/2022 02:17 PM - Luminair

digitalcircuit wrote:

With 0.14/master increasing build requirements, this is something we can finally try to tackle.

It lives! I just got this on Synology DS920+ AMD64 lscr.io/linuxserver/quassel-core:0.14.0-ls100. The system slowed down while downloading a file quickly and then the Quassel container stopped working. I attempted sqlite3 dump and vacuum to no effect. I don't have the skill do attempt digging into the database to attempt a workaround unless you have a precise suggestion :P

```
2022-10-03 09:03:35 [Info ] Restoring previous core state...  
2022-10-03 09:03:35 [Error] SqliteStorage::getBufferInfo(): received more than one Buffer!  
2022-10-03 09:03:35 [Error] Query: "SELECT bufferid, buffertype, groupid\nFROM buffer\nWHERE buffer.n  
etworkid = :networkid AND buffer.userid = :userid AND buffer.buffercname = :buffercname"  
2022-10-03 09:03:35 [Error] bound Values:  
2022-10-03 09:03:35 [Error] 0 :  
2022-10-03 09:03:35 [Error] 1 : 1  
2022-10-03 09:03:35 [Error] 2 : 2  
2022-10-03 09:03:36 [Error] SqliteStorage::getBufferInfo(): received more than one Buffer!  
2022-10-03 09:03:36 [Error] Query: "SELECT bufferid, buffertype, groupid\nFROM buffer\nWHERE buffer.n  
etworkid = :networkid AND buffer.userid = :userid AND buffer.buffercname = :buffercname"  
2022-10-03 09:03:36 [Error] bound Values:  
2022-10-03 09:03:36 [Error] 0 :  
2022-10-03 09:03:36 [Error] 1 : 1  
2022-10-03 09:03:36 [Error] 2 : 2  
2022-10-03 09:03:36 [Error] SqliteStorage::getBufferInfo(): received more than one Buffer!
```

#6 - 10/20/2022 05:54 PM - Luminair

The database got corrupted again after another shutdown. I believe I captured the log before and after the shutdown. I hope this helps!

```
quassel-core  
date stream content
```

```

2022-10-20T04:30:27.149707308Z stderr 2022-10-20 01:30:27 [Info ] Client connected from 172.18.0.1
2022-10-20T04:30:16.079720515Z stderr 2022-10-20 01:30:15 [Info ] Restoring previous core state...
2022-10-20T04:30:16.079672677Z stderr 2022-10-20 01:30:15 [Info ] Listening for GUI clients on IPv4 0.0.
0.0 port 4242 using protocol version 10
2022-10-20T04:30:16.079634649Z stderr 2022-10-20 01:30:15 [Info ] Listening for GUI clients on IPv6 :: p
ort 4242 using protocol version 10
2022-10-20T04:30:16.079603682Z stderr 2022-10-20 01:30:15 [Info ] Database authenticator is ready.
2022-10-20T04:30:16.079567330Z stderr 2022-10-20 01:30:15 [Info ] SQLite storage backend is ready. Schem
a version: 32
2022-10-20T04:30:16.079502954Z stderr 2022-10-20 01:30:15 [Warn ] SslServer: Certificate expired on "Tue
Nov 23 00:35:19 2021 GMT"
2022-10-20T04:30:15.734894230Z stderr 2022-10-20 01:30:15 [Warn ] SslServer: Certificate expired on "Tue
Nov 23 00:35:19 2021 GMT"
2022-10-20T04:30:14.264701984Z stderr s6-rc: info: service 99-ci-service-check successfully started
2022-10-20T04:30:14.264443379Z stdout [ls.io-init] done.
2022-10-20T04:30:14.263123852Z stderr s6-rc: info: service 99-ci-service-check: starting
2022-10-20T04:30:14.262980205Z stderr s6-rc: info: service legacy-services successfully started
2022-10-20T04:30:14.259108462Z stderr s6-services-up: info: copying legacy longrun quassel (no readiness no
tification)
2022-10-20T04:30:14.233464654Z stderr s6-rc: info: service legacy-services: starting
2022-10-20T04:30:14.233361595Z stderr s6-rc: info: service init-services successfully started
2022-10-20T04:30:14.232457828Z stderr s6-rc: info: service init-services: starting
2022-10-20T04:30:14.232350303Z stderr s6-rc: info: service init-mods-end successfully started
2022-10-20T04:30:14.231422907Z stderr s6-rc: info: service init-mods-end: starting
2022-10-20T04:30:14.231261662Z stderr s6-rc: info: service init-mods-package-install successfully starte
d
2022-10-20T04:30:14.224960924Z stderr s6-rc: info: service init-mods-package-install: starting
2022-10-20T04:30:14.224780053Z stderr s6-rc: info: service init-mods successfully started
2022-10-20T04:30:14.223792729Z stderr s6-rc: info: service init-mods: starting
2022-10-20T04:30:14.223670101Z stderr s6-rc: info: service legacy-cont-init successfully started
2022-10-20T04:30:14.223306696Z stderr cont-init: info: /etc/cont-init.d/99-custom-files exited 0
2022-10-20T04:30:14.220354980Z stdout "[custom-init] No custom files found, skipping..."
2022-10-20T04:30:14.216367739Z stderr cont-init: info: running /etc/cont-init.d/99-custom-files
2022-10-20T04:30:14.216278602Z stderr cont-init: info: /etc/cont-init.d/40-config exited 0
2022-10-20T04:30:14.180257486Z stderr cont-init: info: running /etc/cont-init.d/40-config
2022-10-20T04:30:14.180130134Z stderr cont-init: info: /etc/cont-init.d/30-keygen exited 0
2022-10-20T04:30:14.128566537Z stderr cont-init: info: running /etc/cont-init.d/30-keygen
2022-10-20T04:30:14.128419914Z stderr cont-init: info: /etc/cont-init.d/10-adduser exited 0
2022-10-20T04:30:14.075731934Z stdout
2022-10-20T04:30:14.075704657Z stdout
2022-10-20T04:30:14.075659266Z stdout User gid: 101
2022-10-20T04:30:14.075624184Z stdout User uid: 1026
2022-10-20T04:30:14.066379359Z stdout
2022-10-20T04:30:14.066354226Z stdout GID/UID
2022-10-20T04:30:14.066330315Z stdout
2022-10-20T04:30:14.066301743Z stdout https://www.linuxserver.io/donate/
2022-10-20T04:30:14.066270352Z stdout To support LSI0 projects visit:
2022-10-20T04:30:14.066247701Z stdout
2022-10-20T04:30:14.066221796Z stdout
2022-10-20T04:30:14.066193409Z stdout Brought to you by linuxserver.io
2022-10-20T04:30:14.066171165Z stdout
2022-10-20T04:30:14.066147767Z stdout
2022-10-20T04:30:14.066119976Z stdout
2022-10-20T04:30:14.066095482Z stdout |_ | |__/_ | | \__/_/
2022-10-20T04:30:14.066069168Z stdout | | \__ \ | | | ( ) |
2022-10-20T04:30:14.066037550Z stdout | | / __| | | / \
2022-10-20T04:30:14.066010359Z stdout | | _ _ _
2022-10-20T04:30:14.065982946Z stdout _ ( )
2022-10-20T04:30:14.065953424Z stdout
2022-10-20T04:30:14.065743335Z stdout usermod: no changes
2022-10-20T04:30:13.901766797Z stderr cont-init: info: running /etc/cont-init.d/10-adduser
2022-10-20T04:30:13.901720219Z stderr cont-init: info: /etc/cont-init.d/01-migrations exited 0
2022-10-20T04:30:13.901663174Z stdout [migrations] no migrations found
2022-10-20T04:30:13.901572857Z stdout [migrations] started
2022-10-20T04:30:13.899889604Z stderr cont-init: info: running /etc/cont-init.d/01-migrations
2022-10-20T04:30:13.899753554Z stderr cont-init: info: /etc/cont-init.d/01-envfile exited 0
2022-10-20T04:30:13.895966984Z stderr cont-init: info: running /etc/cont-init.d/01-envfile
2022-10-20T04:30:13.893394527Z stderr s6-rc: info: service legacy-cont-init: starting
2022-10-20T04:30:13.893242656Z stderr s6-rc: info: service fix-attns successfully started
2022-10-20T04:30:13.891338332Z stderr s6-rc: info: service 00-legacy successfully started
2022-10-20T04:30:13.888918627Z stderr s6-rc: info: service 00-legacy: starting
2022-10-20T04:30:13.888848199Z stderr s6-rc: info: service fix-attns: starting
2022-10-20T04:30:13.887512397Z stderr s6-rc: info: service s6rc-oneshot-runner successfully started
2022-10-20T04:30:13.881904375Z stderr s6-rc: info: service s6rc-oneshot-runner: starting
2022-10-20T04:30:13.576440657Z stdout "[custom-init] No custom services found, skipping..."

```

```

2022-10-20T04:27:16.784662764Z stderr "2022-10-20 01:27:16 [Info ] Already shutting down, ignoring signal"
2022-10-20T04:27:16.784590324Z stderr 2022-10-20 01:27:16 [Info ] Caught signal 15
2022-10-20T04:27:16.783947570Z stderr s6-rc: info: service s6rc-oneshot-runner successfully stopped
2022-10-20T04:27:16.781596754Z stderr s6-rc: info: service s6rc-oneshot-runner: stopping
2022-10-20T04:27:16.781254844Z stderr s6-rc: info: service fix-attrs successfully stopped
2022-10-20T04:27:16.780415045Z stderr s6-rc: info: service fix-attrs: stopping
2022-10-20T04:27:16.780280357Z stderr s6-rc: info: service legacy-cont-init successfully stopped
2022-10-20T04:27:16.776798598Z stderr s6-rc: info: service legacy-cont-init: stopping
2022-10-20T04:27:16.776728078Z stderr s6-rc: info: service init-mods successfully stopped
2022-10-20T04:27:16.775897494Z stderr s6-rc: info: service init-mods: stopping
2022-10-20T04:27:16.775729911Z stderr s6-rc: info: service init-mods-package-install successfully stopped
2022-10-20T04:27:16.774861581Z stderr s6-rc: info: service init-mods-package-install: stopping
2022-10-20T04:27:16.774716126Z stderr s6-rc: info: service init-mods-end successfully stopped
2022-10-20T04:27:16.773947909Z stderr s6-rc: info: service 00-legacy successfully stopped
2022-10-20T04:27:16.773829099Z stderr s6-rc: info: service init-mods-end: stopping
2022-10-20T04:27:16.773743316Z stderr s6-rc: info: service init-services successfully stopped
2022-10-20T04:27:16.773150924Z stderr s6-rc: info: service 00-legacy: stopping
2022-10-20T04:27:16.772685868Z stderr s6-rc: info: service init-services: stopping
2022-10-20T04:27:16.772573024Z stderr s6-rc: info: service legacy-services successfully stopped
2022-10-20T04:27:16.771725005Z stderr s6-svwait: fatal: timed out
2022-10-20T04:27:13.708305177Z stderr 2022-10-20 01:27:13 [Info ] Core shutting down...
2022-10-20T04:27:13.707406457Z stderr 2022-10-20 01:27:13 [Info ] Quitting...
2022-10-20T04:27:13.707262000Z stderr 2022-10-20 01:27:13 [Info ] Caught signal 15
2022-10-20T04:27:13.694014664Z stderr s6-rc: info: service legacy-services: stopping
2022-10-20T04:27:13.693856469Z stderr s6-rc: info: service 99-ci-service-check successfully stopped
2022-10-20T04:27:13.690303096Z stderr s6-rc: info: service 99-ci-service-check: stopping

```

#7 - 11/24/2022 04:00 AM - Luminair

I've developed some new info about my database failures:

- 1) Docker apparently uses SIGTERM to stop apps, and Quasselcore apparently doesn't handle that, which then leads to a chance of corrupting the database every time Docker closes Quassel. The workaround I'm trying is "stop_signal: SIGINT", equal to a Ctrl+C, which Quassel does handle.
- 2) Quassel database is unpredictably inconsistent and there is no system/API in place to tell it to BECOME consistent before a backup or filesystem snapshot. So this caused my backups to fail, too. The workaround for doing a backup will be to gracefully shut down Quasselcore, do the backup, then start it up again.

Thoughts?