Quassel IRC - Bug #1506

0.13 missing messages

01/01/2019 04:06 PM - galfwender

Status:	Resolved	**Start date:**	01/01/2019
Priority:	High	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
Version:	0.13.0	**OS:**	Linux

Description

Upgraded from v0.12.5 to v0.13 core and faced the following issues. New install of v0.13 core with fresh sqlite db made no difference.

- Messages sent to a channels are visible in Quasseldroid v1.0.5 but not always in v0.13.0 linux client
- Closing and reopening client sometimes fetches messages that were not visible previously, but still not all messages as compared to the view in Quasseldroid
- Quasseldroid will not fetch message backlog for a channel until someone speaks

Associated revisions

Revision 85087a83 - 02/14/2019 08:25 PM - Shane Synan

core: Workaround Qt 4 SQL bindValue() duplicates

Workaround Qt 4 QSqlQuery::bindValue() not handling duplicate
parameter names by giving every duplicated parameter a unique name.

Qt 5 handles this as one expects, so this doesn't need forward-ported
to 0.14/master where Qt 4 support has been dropped.

Applies to SQLite and PostgreSQL.

Note: PostgreSQL prepared statements properly handle repeated "$"
parameters, so there's no need to modify those.

Test case for prepared statements:

```
PREPARE qcore_test_var

AS SELECT * FROM buffer WHERE bufferid = $1 AND userid = $1
AND networkid = $1;

EXECUTE qcore_test_var(1);
```

Gives a result if bufferid = userid = networkid = 1 exists.

Credit to @justJanne for finding the root cause of the issue, and
suggesting the fix, and to 'galfwender' and @darkstar for reporting.

Fixes #1506

See https://doc.qt.io/archives/qt-4.8/qsqlquery.html#bindValue
And https://doc.qt.io/qt-5/qsqlquery.html#bindValue

History

#1 - 01/01/2019 09:48 PM - galfwender

Cannot replicate the issue when using v0.13 linux client with v0.12.5 core

#2 - 01/22/2019 08:30 PM - Darkstar

- File quassel_1.jpg added

- File quassel_2.jpg added

Can confirm with Windows client. The priority should be raised since this is pretty much breaking Quassel for everyday use.

Restarting the core brings back some messages (so they are definitely in the database and not lost), however not all

Two images to illustrate this:

- Image 1 shows the missing messages in the Windows client (they are in the Android client, but only after someone wrote in the channel)
  quassel_1.jpg
- Image 2 shows the empty Android client before someone posts in the channel
  quassel_2.jpg

#3 - 01/26/2019 06:43 AM - digitalcircuit

- Priority changed from Normal to High

This seems like an issue with the performance fix for fetching unread backlog messages… There were a series of pull requests - #273, #274, #276 - aimed at improving performance for backlog fetching when really old buffers existed.

For those affected, what database backend are you using - SQLite or PostgreSQL?

Does inspecting the database for buffer.lastmsgid…

```
SELECT lastmsgid FROM buffer WHERE bufferid = buffer_ID_here;
```

…and comparing it with the real value…

```
SELECT messageid FROM backlog WHERE bufferid = buffer_ID_here ORDER BY messageid DESC LIMIT 1;
```

…result in different numbers?

To get the buffer ID, you can run…

```
SELECT bufferid, buffername FROM buffer WHERE buffername = 'buffer_name_here';
```

…removing the WHERE clause if you want to get a whole list.

Reasoning:

This is tracking down whether or not the database is storing the right lastmsgid numbers. If so, Quassel's query to fetch messages may be broken. If not, the database might not properly be running the trigger to update the lastmsgid.

#4 - 01/28/2019 11:52 AM - Darkstar

I am using an SQLite DB as a backend.

I tried the SQL commands for two buffers (those from the screenshots above), and the numbers seem to match:

```
root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT lastmsgid FROM buffer WHERE bufferid = 94
;"
5924458
root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT messageid FROM backlog WHERE bufferid = 9
4 ORDER BY messageid DESC LIMIT 1;"
5924458
root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT lastmsgid FROM buffer WHERE bufferid = 13
2;"
5924700
root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT messageid FROM backlog WHERE bufferid = 1
32 ORDER BY messageid DESC LIMIT 1;"
5924700
```

#5 - 01/29/2019 02:03 AM - digitalcircuit

Thank you for the feedback! This seems to confirm it's an issue with the fetching of backlog, not the tracking of newest messages (which could also result in missing messages).

I've gotten in touch with the Quasseldroid developer justJanne over on Freenode/#quassel, confirmed that others experience this issue too, and we'll look into this within the next few days (if time permits).

#6 - 01/29/2019 09:24 AM - Darkstar

Thanks! Again, this also happens for the Windows client (and, as the OP reported, the Linux client) so it's not limited to Quasseldroid.

#7 - 01/31/2019 11:51 AM - justjanne

Hi, while trying to debug this, I'd like to verify if there's something wrong with the database itself, or the queries. To ensure that this is not the case, I'd like to have you run the following queries:

```
SELECT bufferid, networkid, buffertype, groupid, buffername FROM buffer WHERE bufferid = buffer_ID_here AND
 userid = user_ID_here
```

```
SELECT COUNT(*) FROM backlog JOIN sender ON backlog.senderid = sender.senderid WHERE
 bufferid = buffer_ID_here AND backlog.messageid <= (SELECT buffer.lastmsgid FROM buffer WHERE
 buffer.bufferid = buffer_ID_here) ORDER BY messageid DESC LIMIT 150
```

To do this, you'll need the bufferid (which you've already found earlier) and the userid, which you can get with this query:

```
SELECT userid FROM quasseluser WHERE username = 'user_name_here'
```

If these queries return proper results as well, we'd need to do some more complicated testing, but it's possible that at some point your SQL Indexes just got corrupted, which could also lead to issues like these (although it is more probable that it's a core-side issue).

#8 - 01/31/2019 11:58 AM - justjanne

Additionally, please check if your core has been logging any errors when a client connects.

Note for other developers:

A potential candidate for the issue could be if users have many networks, and the core is used by many users, the following function:

```
bool SqliteStorage::safeExec(QSqlQuery& query, int retryCount)
{
    query.exec();

    if (!query.lastError().isValid())
        return true;

    switch (query.lastError().number()) {
    case 5:  // SQLITE_BUSY         5   /* The database file is locked */
             // fallthrough
    case 6:  // SQLITE_LOCKED       6   /* A table in the database is locked */
        if (retryCount < _maxRetryCount)
            return safeExec(query, retryCount + 1);
        break;
    default:
        ;
    }
    return false;
}
```

_maxRetryCount is set to 150, but it's possible that your core has reached that limit.

I'd be very worried if that function is the cause of the trouble, though, as it's been basically unmodified since 2008.

#9 - 01/31/2019 12:50 PM - Darkstar

Hi,

Since I'm the only user using this Quassel core, my user ID is 1 (this might also rule out the "if the core is used by many users" in the second comment). Here are the results for the same two buffers used above:

```
root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT userid FROM quasseluser;"
1
root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT bufferid, networkid, buffertype, groupid,
 buffername FROM buffer WHERE bufferid = 94 AND userid = 1;"
94|1|2||#DiscImageChef

root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT COUNT(*) FROM backlog JOIN sender ON back
log.senderid = sender.senderid WHERE bufferid = 94 AND backlog.messageid <= (SELECT buffer.lastmsgid FROM buff
er WHERE buffer.bufferid = 94) ORDER BY messageid DESC LIMIT 150"
10152

root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT bufferid, networkid, buffertype, groupid,
 buffername FROM buffer WHERE bufferid = 132 AND userid = 1;"
132|1|2||#dosbox

root@server:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT COUNT(*) FROM backlog JOIN sender ON back
log.senderid = sender.senderid WHERE bufferid = 132 AND backlog.messageid <= (SELECT buffer.lastmsgid FROM buf
fer WHERE buffer.bufferid = 132) ORDER BY messageid DESC LIMIT 150"
28303
```

I don't know if these numbers are "sensible" or not though, since I'm not that familiar with the DB schema (and it's been a few years since I last had to write my own SQL queries ;-)

For completeness, here's again the numbers of the first two queries, in case they have changed in the last few days (probably) and in case they are somehow related to these new queries/results:

```
root@h2387211:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT messageid FROM backlog WHERE bufferid =
 94 ORDER BY messageid DESC LIMIT 1;"
5937766

root@h2387211:/var/lib/quassel# sqlite3 quassel-storage.sqlite "SELECT messageid FROM backlog WHERE bufferid =
 132 ORDER BY messageid DESC LIMIT 1;"
5937776
```

As for errors, I have nothing more than Warnings in my log file, and those are in there since 2017 (expired SSL certificate) so I don't think it mattesrs much. Still, here's a copy (I restarted the core once, that's why the shutdown message is in there)

```
2019-01-30 00:08:31 [Info ] Client ::FFFF:xxxx:xxxx%0 disconnected (UserId: 1).
2019-01-30 20:25:55 [Info ] Client connected from ::FFFF:xxxx:xxxx%0
2019-01-30 20:25:55 [Info ] Client ::FFFF:xxxx:xxxx%0 initialized and authenticated successfully as "Darkstar"
 (UserId: 1).
2019-01-30 20:25:55 [Info ] Client supports unknown features: EcdsaCertfpKeys
2019-01-30 23:17:49 [Info ] Client ::FFFF:xxxx:xxxx%0 disconnected (UserId: 1).
2019-01-31 12:03:44 [Info ] Caught signal 15
2019-01-31 12:03:44 [Info ] Quitting...
2019-01-31 12:03:44 [Info ] Core shutting down...
2019-01-31 12:03:45 [Info ] Core shutdown complete!
2019-01-31 12:37:21 [Warn ] SslServer: Certificate expired on Sat Sep 9 12:08:48 2017
2019-01-31 12:37:21 [Warn ] SslServer: Certificate expired on Sat Sep 9 12:08:48 2017
2019-01-31 12:37:21 [Warn ] PostgreSQL driver plugin not available for Qt. Installed drivers: QSQLITE, QMYSQL3
, QMYSQL
2019-01-31 12:37:21 [Info ] SQLite storage backend is ready. Schema version: 31
2019-01-31 12:37:21 [Info ] Database authenticator is ready.
2019-01-31 12:37:21 [Info ] Listening for GUI clients on IPv6 :: port 4242 using protocol version 10
2019-01-31 12:37:21 [Info ] Restoring previous core state...
2019-01-31 12:41:01 [Warn ] "Timed out quitting network EFnet (network ID: 5, user ID: 1)"
2019-01-31 12:44:41 [Warn ] "Timed out quitting network EFnet (network ID: 5, user ID: 1)"
```

#10 - 01/31/2019 05:26 PM - justjanne

Thanks, that rules out any issues with the database or the storage layer itself, and means it's somewhere either the CoreBacklogManager or the SqliteStorage, which means the task is much harder than I had hoped for it to be.

#11 - 02/06/2019 12:56 AM - justjanne

It's a Qt bug
Qt4 on Ubuntu 14.04 and related debian versions is broken
QSqlQuery can't handle the same named parameter appearing in multiple places

```
SELECT * FROM backlog WHERE messageid <= :messageid AND messageid >= :messageid
```

returns nothing.

```
SELECT * FROM backlog WHERE messageid = :messageid
```

works for the same value

This leads to an issue with select_messagesNewerThan.sql and all filtered queries.

Patching select_messagesNewerThan to

```
SELECT messageid, time,  type, flags, sender, senderprefixes, realname, avatarurl, message
FROM backlog
JOIN sender ON backlog.senderid = sender.senderid
WHERE backlog.messageid >= :firstmsg
    AND backlog.messageid <= (SELECT buffer.lastmsgid FROM buffer WHERE buffer.bufferid = :bufferid2)
    AND bufferid = :bufferid
ORDER BY messageid DESC
LIMIT :limit
```

while also patching the code to accordingly set the bufferid2 parameter as well works.

This issue appears for select_messagesAll_filtered.sql, select_messagesAllNew_filtered.sql, select_messagesNewerThan_filtered.sql,

select_messagesNewestK_filtered.sql, select_messagesRange_filtered.sql and select_messagesNewerThan.sql. Therefore a different, more thorough solution than this simple patch is needed.

#12 - 02/06/2019 01:14 AM - Darkstar

I'm indeed using Ubuntu 14.04.5 on that server (it's a virtual server so I can't update whenever I want to, sadly).

Here are the versions I'm running, maybe we can narrow it down.

```
ii  libsqlite3-0:amd64              3.8.2-1ubuntu2.1                          amd64        SQLite 3 shared li
brary
ii  sqlite3                         3.8.2-1ubuntu2.1                          amd64        Command line inter
face for SQLite 3
ii  libqt4-network:amd64            4:4.8.5+git192-g085f851+dfsg-2ubuntu4.1   amd64        Qt 4 network modul
e
ii  libqt4-sql:amd64                4:4.8.5+git192-g085f851+dfsg-2ubuntu4.1   amd64        Qt 4 SQL module
ii  libqt4-sql-mysql:amd64          4:4.8.5+git192-g085f851+dfsg-2ubuntu4.1   amd64        Qt 4 MySQL databas
e driver
ii  libqt4-sql-sqlite:amd64         4:4.8.5+git192-g085f851+dfsg-2ubuntu4.1   amd64        Qt 4 SQLite 3 data
base driver
ii  libqtcore4:amd64                4:4.8.5+git192-g085f851+dfsg-2ubuntu4.1   amd64        Qt 4 core module
ii  qtcore4-l10n                    4:4.8.5+git192-g085f851+dfsg-2ubuntu4.1   all          Qt 4 core module t
ranslations
ii  quassel-core                    0.13.0-0ubuntu0~ppa0~14.04.0              amd64        distributed, KDE/Q
t-based IRC client - core/server component
```

#13 - 02/06/2019 01:31 AM - digitalcircuit

This a known/intentional Qt 4 limitation... https://doc.qt.io/archives/qt-4.8/qsqlquery.html#bindValue

Values cannot be bound to multiple locations in the query, eg:

```
INSERT INTO testtable (id, name, samename) VALUES (:id, :name, :name)
```

Binding to name will bind to the first :name, but not the second.

Qt 5 handles it. https://doc.qt.io/qt-5/qsqlquery.html#bindValue

This only needs fixed in Qt 4, but I'll likely make the patch for 0.14 and backport just in case.

#14 - 02/06/2019 01:36 AM - digitalcircuit

Additionally, it's possible that Qt 4 can just add in redundant bindValue() calls as needed, within a QT_VERSION < QT4's number block. I'll try this approach first (less code changes == likely less risk of new breakage).

#15 - 02/06/2019 08:58 AM - digitalcircuit

Work-in-progress patch: [0.13] core: Workaround Qt 4 SQL bindValue() duplicates

I still need to test this, write a description, and such, but in the meantime, it's possible to try building Quassel 0.13 on Qt 4 from source to see if this fixes the issue.

#16 - 02/06/2019 11:49 AM - Darkstar

- File quassel_3.png added

I took that branch, compiled quasselcore from it against Qt4, and it indeed seems to work. At least I see all the "day changed to ..." messages that were previously missing, and also the backlog from the above screenshots is there if I scroll up far enough (see image)

thanks for the quick fix, even if it's still only a WIP patch ;-)

#17 - 02/06/2019 11:50 AM - Darkstar

Screenshot:

```
[20:21:41]      <claunia> yeah
[20:21:46]   <Dark-Star> he apparently has multiple
[20:21:47]      <claunia> xD
[20:21:54]   <Dark-Star> "The drive in the pictures has been opened to take photos but you will receive
[20:22:05]   <Dark-Star> ask him to keep the opened one for you to buy later :)
[20:22:07]      <claunia> deposits and estate agencies here are really really expensive
[20:22:17]      <claunia> I had to ask for a loan in the job
[22:37:34]      <claunia> ok so next thing to do....
[22:37:57]      <claunia> correct JaguarCD offsets
[23:05:51]      <claunia> fixing speed calculation...
[00:00:00]                - {Day changed to Dienstag, 22. Januar 2019}
[00:28:55]        <VGPC> <Schrodinger [Brazil]> I have myst and myst demo if you need.
[18:58:30]      <claunia> @Schrodinger Jaguar CD?
[18:59:06]        <VGPC> <Schrodinger [Brazil]> Yes
[19:20:17]      <claunia> can you create a dicf image of both with current version in the dropbox link?
[19:33:15]        <VGPC> <Schrodinger [Brazil]> By the weekend, yes
[20:07:44]      <claunia> good
```

#18 - 02/14/2019 08:25 PM - Anonymous

- Status changed from New to Resolved

- % Done changed from 0 to 100

Applied in changeset quassel|85087a839d89dc53ed71b41e12244e57f21f67b1.

Files

quassel_1.jpg	481 KB	01/22/2019	Darkstar
quassel_2.jpg	353 KB	01/22/2019	Darkstar
quassel_3.png	12.6 KB	02/06/2019	Darkstar