OsmoNITB - Bug #1591

libdbi is buggy and slow, get rid of it

02/23/2016 03:48 PM - laforge

Status:	Closed	**Start date:**	02/23/2016
Priority:	Low	**Due date:**	
Assignee:	neels	**% Done:**	80%
Category:			
Target version:			
Spec Reference:			

Description

libdbi seemed like a good idea at the time, but in reality we always only used the sqlite3 backend, and libdbi causes more trouble than advantages. Both in terms of bugs, as well as performance. We should simply use sqlite3 directly.

And of coure move to an asynchronous database interface, too - but that's a separate topic.

Related issues:

Related to OsmoNITB - Feature #1592: VLR in libmsc, to connect to HLR asynchr...	**Closed**	**02/23/2016**
Related to OpenBSC - Bug #30: sqlite3 database / asynchronous access to it	**Closed**	
Related to Cellular Network Infrastructure - Feature #1860: Include Ubuntu 16...	**Closed**	**11/28/2016**
Related to Cellular Network Infrastructure - Support #1923: re-enable all nig...	**Rejected**	**01/19/2017**

History

#1 - 02/23/2016 03:48 PM - laforge

migrated to https://projects.osmocom.org/issues/1591

#2 - 03/17/2016 09:55 AM - laforge

- Assignee set to laforge

#3 - 04/28/2016 07:11 PM - laforge

- Status changed from New to In Progress

#4 - 04/28/2016 07:11 PM - laforge

- Related to Feature #1592: VLR in libmsc, to connect to HLR asynchronously added

#5 - 05/09/2016 07:38 PM - laforge

- Priority changed from High to Urgent

#6 - 11/11/2016 08:38 AM - laforge

- Assignee changed from laforge to neels

#7 - 12/02/2016 05:17 PM - laforge

- Target version set to Asynchronous HLR+AUC for CS

#8 - 12/09/2016 12:43 AM - neels

I take it that once libvlr is in place, there will be no libdbi in osmo-nitb/osmo-cscn.
-> watch #1592

#9 - 01/04/2017 04:11 PM - neels

I'm noticing that we apparently do still need the db for SMS.
So what was previously the hlr.sqlite3 database is mostly replaced by libvlr,
but it will remain as an sms.sqlite3.

Am I missing something?

#10 - 01/05/2017 09:16 AM - laforge

On Wed, Jan 04, 2017 at 04:11:00PM +0000, neels [REDMINE] wrote:

> I'm noticing that we apparently do still need the db for SMS.
> So what was previously the hlr.sqlite3 database is mostly replaced by libvlr,
> but it will remain as an sms.sqlite3.

yes, until the SMSC has been externalized [if needed] at some later
point. It could e.g. be comparable to MNCC: A run-time option whether
to run with the internal handler, or to use SMPP in transaction mode
only without any internal store-and-forward?

From past experience: The SMS load in practise has not been that much of
an issue compared to the "HLR" load.

--
- Harald Welte <laforge@gnumonks.org> http://laforge.gnumonks.org/
==========
"Privacy in residential applications is a desirable marketing option."
(ETSI EN 300 175-7 Ch. A6)

#11 - 01/05/2017 02:01 PM - neels

- Related to Bug #30: sqlite3 database / asynchronous access to it added

#12 - 01/18/2017 02:05 PM - neels

- Status changed from In Progress to New

#13 - 01/19/2017 02:41 PM - neels

- Blocks Support #1923: re-enable all nightly package builds added

#14 - 01/19/2017 11:54 PM - stuge

I started looking into this several years ago and I'd still like to help if I can.

Maybe I can't help code within your required timeframe, but I would love to at least have a brainstorming/design discussion - ideally in person, but online works too.

I can contribute experience with SQLite, MySQL/MariaDB and PostgreSQL in C and SQL-fu.

#15 - 01/20/2017 04:12 PM - msuraev

- Related to Feature #1860: Include Ubuntu 16.04 LTS in nightly builds added

#16 - 01/20/2017 10:56 PM - neels

Hi stuge!

Help is always welcome.

This change must be seen in the light of #1592, IOW the neels/vlr branch (based on laforge's VLR work) that pretty much replaces half of our MSC with a brand new VLR implementation and an external OsmoHLR process.

The point is that the sqlite file that OsmoNITB keeps will not store anything about subscribers anymore. It will only be used for SMS, if at all.

Let me paste a relevant snippet from #1592:

> With SMS, the problem is still the legacy db. Obtaining an SMS from the DB entails a JOIN with the no longer
> populated Subscriber table (to pick only those that are active, i.e. lac >= 0).
> I'll have to rethink the SMS code.
>
> How to deal with SMS that have no active subscriber: in the long term we will need a plan
> for undelivered SMS [whose recipients] show up with a different VLR, so more of a quick hack is
> probably sufficient [for local delivery] at this point, as long as we assume that there is only one VLR.
>
> The SMS code should also be moved away from libdbi to native sqlite.
> I will see whether doing both changes in one go makes sense.
> The database file should probably also be renamed to sms.db.

Questions to clarify:

- what to do with SMS in general? I'm not at all familiar with SMS delivery beyond the simple case where sender and recipient are both actively available at the same OsmoNITB. What's the roadmap? Do we even *need* SMS storage in a DB, or can we live with an in-memory llist storage until the bigger picture resolves?
- Given that we still need to keep SMS in a DB in OsmoNITB, we need to adjust the code that looks for pending SMS. So far it could SELECT only those SMS with a recipient that is actively attached (by querying "lac > 0"). Now we'd have to SELECT one or more SMS and in a second step look up whether their recipients are currently active in the VLR. Not hard, but harder to make efficient.
- Finally, given that we keep SMS in a DB, and while adjusting the SQL anyway, we could also move away from libdbi in one fell swoop. <-- and this point is what this issue is about.

If you have any ideas or patches for the openbsc.git:neels/vlr branch, they would be very welcome.

#17 - 01/23/2017 01:15 PM - laforge

On Fri, Jan 20, 2017 at 10:56:48PM +0000, neels [REDMINE] wrote:

> - what to do with SMS in general? I'm not at all familiar with SMS delivery beyond the simple case where sender and recipient are both actively available at the same OsmoNITB. What's the roadmap?

See below.

> Do we even *need* SMS storage in a DB, or can we live with an in-memory llist storage until the bigger picture resolves?

no, the latter is clearly not sufficient. A persistent database is needed.

However, that database should sooner or later not be part of the NITB as the NITB is moving towards becoming a classic VLR+MSC only. At that point, all SMS should be handled by an external process.

In general I would like to see SMPP to be used here, but unfortunately there is somewhat of an "impedacne mis-match" between what is required and what SMPP is designed for: SMPP is intended for communication of an SMSC with external ESMEs (sms receiving/sending applications), and not for the communication between the MSC and the SMSC.

The Communication between MSC and SMSC (which is what we're looking at here) is normally done over MAP, using procedures like MO-forward-SM to the GT of the SMSC. It's thus on the same level as GSUP.

So the question for the mid to long term is:

- do we want to use SMPP for the MSC-SMSC interface despite its inadequacies, maybe with some custom extensions?
- do we want to introduce a new protocol?
- do we want to hack up GSUP to include this? (probably not)

Please also note that Fairwaves was working on something in this area which they then used to convert SMS to SIP, AFAIR.

> - Given that we still need to keep SMS in a DB in OsmoNITB, we need to adjust the code that looks for pending SMS. So far it could SELECT only those SMS with a recipient that is actively attached (by querying "lac > 0"). Now we'd have to SELECT one or more SMS and in a second step look up whether their recipients are currently active in the VLR. Not hard, but harder to make efficient.

I wouldn't worry about efficiency at this point.

The short term goal is to get the HLR/VLR split done and spend as little as possible on side-track issues like SMS.

> - Finally, given that we keep SMS in a DB, and while adjusting the SQL anyway, we could also move away from libdbi in one fell swoop. <-- and this point is what this issue is about.

I am wondering if that intermediate step is really useful. The question is how long the sms handling will remain within the NITB, before being externalized.

#18 - 02/02/2017 04:11 PM - neels

I think I will implement lookup code that finds pending SMS for active subscribers
by looking in the VLR for active subscribers (RAM) and matching that up with pending
SMS recipients (SQL), to get basic functionality disregarding efficiency. (later)

#19 - 02/20/2017 01:22 PM - neels

(on neels/vlr branch, SMS lookup is now done by round-robin on receiver MSISDN by SQL in the SMS db
and per hit a lookup in the VLR subscribers list in RAM)

#20 - 03/10/2017 10:10 PM - laforge

- % Done changed from 0 to 80

#21 - 09/15/2017 12:24 PM - keith

laforge wrote:

> Please also note that Fairwaves was working on something in this area
> which they then used to convert SMS to SIP, AFAIR.

Could anybody from fairwaves point to branch/commits related to this?

#22 - 09/16/2017 05:39 PM - ipse

keith wrote:

> laforge wrote:
>
> > Please also note that Fairwaves was working on something in this area
> > which they then used to convert SMS to SIP, AFAIR.
>
> Could anybody from fairwaves point to branch/commits related to this?

See a discussion on the mailing list about external USSD. We have only one active branch (per repo) which has all our changes in it. So it's the same branch with both external USSD, SMS, and LURs.

I think we should create a separate ticket for the external SMS functionality like this one for external USSD support: https://osmocom.org/issues/1597

#23 - 09/27/2017 10:06 PM - neels

- Priority changed from Urgent to Normal

#24 - 10/29/2017 06:33 PM - laforge

- Priority changed from Normal to Low

#25 - 10/29/2017 06:52 PM - laforge

- Target version deleted (Asynchronous HLR+AUC for CS)

#26 - 12/03/2017 11:02 AM - laforge

- Blocks deleted (Support #1923: re-enable all nightly package builds)

#27 - 12/03/2017 11:02 AM - laforge

- Related to Support #1923: re-enable all nightly package builds added

#28 - 03/05/2018 07:35 PM - laforge

- Status changed from New to Closed