

Redmine - Defect #13210

500 with sqlite exception (database locked)

2013-02-18 14:49 - Dietmar H

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Database	Estimated time:	0.00 hour
Target version:		Affected version:	2.2.3
Resolution:	Invalid		

Description

After moving our Redmine installation to another server (slower HW then before, but in Intranet) and updating it from 1.3 to 2.2.3, every few days it stops working because of locked DB. Restarting Apache solves the problem. Usually the problems starts with a log entry

```
ActiveRecord::StatementInvalid (SQLite3::BusyException: cannot rollback transaction - SQL statements in progress: rollback transaction):
```

After that, every write access results in

```
ActiveRecord::StatementInvalid (SQLite3::BusyException: database is locked: commit transaction):
```

Before the server and version migration we used it for about 1 year, with comparable concurrency (~ 10 team members) and this didn't happen once.

I already tried adding

```
timeout: 10000
```

to database.yml with no luck.

crontabs installed are:

```
31 * * * * cd /usr/share/redmine && ./script/rails runner "Repository.fetch_changesets" -e production
01 7 * * 1-4 cd /usr/share/redmine && rake redmine:send_reminders days=7 RAILS_ENV="production" > /dev/null
```

I know moving to a "real" DB would solve this problem, but I would prefer avoiding that, since migration wouldn't be a trivial issue and sqlite has it's advantages (e.g. very easy to backup). I think in our case it may even be faster then with MySQL.

Any suggestions are welcome.

Related issues:

Related to Redmine - Defect #15827: Redmine is using a buggy Rails SQLite ada...

Closed

History

#1 - 2013-02-18 15:45 - Jan Niggemann (redmine.org team member)

Related to [#2775](#)?

SQLite will lock the entire file for writing, i.e. during the time it takes for data to be written to disk, nothing else can access the DB ==> no concurrent writes.

#2 - 2013-02-18 16:13 - Etienne Massip

- Status changed from New to Closed

- Resolution set to Invalid

Indeed.

Don't use SQLite in production, it isn't meant for concurrent (multi-user) use.

#3 - 2013-02-19 13:55 - Dietmar H

I see. This seems to have changed, in the official doc for 1.3 it wasn't discouraged.

It's true that SQLite needs full file locks when writing, however several tests show that the level of concurrency needs to be quite high in order for performance to seriously suffer. For small to medium teams it does perfectly fine (and as I stated, it did with Redmine 1.3).

It's a pity that support was dropped, however I understand such decisions, as every supported option takes resources. Looks like our migration to 2.x gets more expensive than planned...

#4 - 2013-02-19 17:36 - Etienne Massip

Dietmar Hofer wrote:

It's a pity that support was dropped

It is not, SQLite is supported, just discouraged.

I don't know how to have it working smoothly with Redmine 2/Rails 3 in a multi-user environment but since it used to work with 1.3 I guess it might be possible?

#5 - 2013-02-19 18:31 - Dietmar H

Etienne Massip wrote:

I don't know how to have it working smoothly with Redmine 2/Rails 3 in a multi-user environment but since it used to work with 1.3 I guess it might be possible?

I also think it's possible. According to <http://stackoverflow.com/questions/78801/sqlite3busyexception> the way to deal with sqlite locks is to shortly wait and retry. I don't know if that was the case in 1.3, but since we didn't have the issue in a year running 1.3 with the same concurrency, I imagine something was different.

#6 - 2013-02-19 19:04 - Dietmar H

See [Moving from sqlite to mysql: plugin migration fails](#) for how to migrate.

#7 - 2014-01-07 19:39 - Timur Alperovich

The errors above are not actually concurrency issues with SQLite3. They are bugs in the way SQLite3 is used.

The original bug posted also indicates an error in the way the adapter is handling concurrent access. Subsequent "database is locked" messages are explained here:

<http://www.sqlite.org/cvstrac/wiki?p=DatabasesIsLocked>

The gist is that something **really** bad happened in terms of using the given connection and it is most certainly an issue in the use of the database (most likely in the adapter layer).

A concurrency error looks exactly like the second one posted on the bug -- the database is locked by a process (you can typically find out which one, at least on Linux, by looking in /proc/locks and looking up the PID for the lock holding the DB open). But I strongly suspect this is an actual bug.

SQLite3 adapter in Rails 3.2.xx has the following bug as well: <https://github.com/rails/rails/issues/13595>

#8 - 2014-01-08 22:54 - Timur Alperovich

Most likely, the problem in this bug is a result of <https://github.com/rails/rails/issues/13631> (again, a bug in the ActiveRecord SQLite3 adapter, which is still present in Rails 4).

#9 - 2014-01-15 11:03 - Toshi MARUYAMA

- Related to Defect #15827: Redmine is using a buggy Rails SQLite adapter added