Redmine - Defect #39437

MySQL / MariaDB issue nested set deadlocks and consistency

2023-10-30 10:54 - Jens Krämer

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marius BĂLTEANU	**% Done:**	0%
Category:	Database	**Estimated time:**	0.00 hour
Target version:	5.1.1		
Resolution:	Fixed	**Affected version:**	

Description

This relates to #19344 #19395 #23318 and #35014.

I did some investigation into this topic for Planio and thought I'd share my findings.

All tests were done on Linux, with Ruby 3.1 and Redmine master, databases (MySQL 5.6, 5.7 and MariaDB 11) in Docker.

To fix the deadlocks, I added a global lock, basically serializing all nested set modifications. I chose to use
select id from settings for update for this because it's a relatively static and small table, but in the
end it could be anything. The changed locking SQL from #23318-18 alone still resulted in deadlocks 100% of the time
in my setup, but since it was determined to be faster back then and the necessary mysql-specific branch was in the code now I added it anyway.

This is certainly neither elegant nor ideal, however I wanted to get around the deadlocks to reproduce another problem we were seeing in the wild - corrupted nested sets, which now indeed happened along with stale record errors with every test run on all three databases.

In my understanding, this is because of MySQLs default transaction isolation level, which is REPEATABLE READ (many
(most?) other commonly used RDBMS default to READ COMMITTED). The relevant difference between the two is, that in the
latter your transaction "sees" changes committed by other transactions as soon as they are COMMITed, while with REPEATABLE READ,
you continue to work on a snapshot that was created once, at the beginning of your transaction.
See i.e. https://mariadb.com/kb/en/set-transaction/#isolation-levels for reference.

In our case (concurrent modifications to the issue nested set, i.e. by parallel modification of various issues'
parent_id), this means:

- the transaction is implicitly opened by Rails when Issue#save is called
- the nested set is locked at a later point in time (in an after_save hook)
- this time gap between snapshot creation (begin of transaction) and actual locking leaves room for race conditions.

A failing sequence of events might look like this (assuming the modified issues are part of the same (larger)
nested set):

- A starts transaction to update issue X
- B starts transaction to update issue Y
- several SELECTs are made by A and B (during validations etc), each now work on their own snapshot, each not seeing modifications made by the other
- B locks the nested set
- A attempts to lock the same, has to wait
- B modifies the set and commits, giving up the locks.
- A now gets the lock, but still has the old snapshot, so it doesn't see the changes made by B
- A now works on the set with stale data, which may lead to a corrupted tree and stale record errors

To fix this, the second patch adds an initializer that sets the tx isolation to "READ COMMITTED" by hooking into
the MySQL2 adaptor. With this in place, all tests now pass all the time, with MySQL 5.6, 5.7 and MariaDB 11.

Now, I am not 100% sure if we should put either of these patches into Redmine, but since it's, right now, not easy to work around the deadlocks by retrying the transaction (that would mean retrying the Issue#save, which may or may not be free of unwanted side effects), I do not see many alternatives if we want to fix this. Maybe we introduce a dedicated table
with a dedicated row that serves as the mutex to make it look a bit less "hacky".

Relaxing the TX isolation level might have side effects (although none appear to be discovered by our current test suite), and one might consider it bad practice to do so at the application level, at all. So probably this initializer and the reasoning behind it should just go into the Wiki as part of MySQL/MariaDB specific setup instructions. After all, the same effect can be achieved by simply setting the default TX isolation level in the database server config. In theory, switching to a lower tx isolation level should help reduce the probability of deadlocks, but this alone at least did not make the concurrency test pass in my setup.

Another possible solution / workaround would be to move the whole nested set manipulation to a separate transaction. This could be done by moving it into an after_commit hook and opening a new transaction in which the snapshot is created with the SELECT ... FOR UPDATE. This way it should work regardless of the configured TX isolation level. With such a separate (and much smaller) transaction, any deadlocks might also easily be handled by retrying the transaction, instead of using the global lock workaround I introduced above.

Related issues:

Related to Redmine - Defect #19344: MySQL 5.6: IssueNestedSetConcurrencyTest#...	**Closed**
Related to Redmine - Feature #19395: Support MariaDB	**New**
Related to Redmine - Defect #23318: #lock_nested_set very slow on mysql with ...	**Closed**
Related to Redmine - Feature #35014: Review and update supported database eng...	**Closed**
Related to Redmine - Feature #35685: Support for MySQL > 5.7 or MariaDB	**Closed**

Associated revisions

Revision 22458 - 2023-11-18 15:39 - Marius BĂLTEANU

Use a global lock provided by with_advisory_lock gem to work around deadlock issues when MySQL >= 5.7 (#39437).

Patch by Jens Krämer.

Revision 22459 - 2023-11-18 15:40 - Marius BĂLTEANU

Moves create parent issue journal to after_commit hook to work around stale object errors on concurrency (#39437).

Patch by Jens Krämer.

Revision 22460 - 2023-11-18 17:36 - Marius BĂLTEANU

Revert undesired change (#22458).

Revision 22461 - 2023-11-18 23:27 - Marius BĂLTEANU

Add concurrent subtask removal test to cover corrupted nested sets (#39437).

Patch by Jens Krämer.

Revision 22462 - 2023-11-18 23:30 - Marius BĂLTEANU

Workaround to use READ-COMMITTED as transaction_isolation level when running the concurrency tests in MySQL. (#39437).

Revision 22464 - 2023-11-19 08:10 - Marius BĂLTEANU

Use tx_isolation for MySQL lower than 8. (#39437).

Revision 22467 - 2023-11-19 12:02 - Marius BĂLTEANU

Merged r22458, r22459, r22460, r22461, r22462 and r22464 from trunk to 5.1-stable (#39437).

History

#1 - 2023-10-31 00:16 - Marius BĂLTEANU

- Target version set to 6.0.0

Thanks for the detailed investigation, I'm setting the target version to 6.0.0 because I think it is a good moment to conclude on this issue.

Regarding the patches, I'm on the same page with you, I'm not sure if should enforce the transaction isolation level from Redmine, but we could recommend this in the installation docs and maybe strongly recommend PostgreSQL as database (1).

I was able to fix the concurrency_test with your first patch by using a begin rescue in create_or_update, but the test added by you fails. Applying both patches worked for me as well using MySQL 8.

Maybe it's better to explore:
2) your second option (with after_commit)

3) try to see which other gems are available the implements this hierarchy and what means a migration to it:

- typed_dag - it looks like is maintained by OpenProject (another Redmine fork?)
- closure_tree - looks like very well maintained.

4) take into consideration some delayed job? but being quite a sensitive action, I'm not sure if it's suitable.

Anyway, you have my full support on this one.

#2 - 2023-11-01 00:54 - Go MAEDA

- Related to Defect #19344: MySQL 5.6: IssueNestedSetConcurrencyTest#test_concurrency : always fails added

- Related to Feature #19395: Support MariaDB added

- Related to Defect #23318: #lock_nested_set very slow on mysql with thousands of subtasks added

- Related to Feature #35014: Review and update supported database engines and versions added

#3 - 2023-11-01 08:33 - Jens Krämer

- File 0001-use-a-global-lock-to-work-around-deadlock-errors-wit.patch added

- File 0002-moves-create_parent_issue_journal-to-after_commit.patch added

Hi Marius, thank you for looking into this!

From what I can tell `typed_dag` supports MySQL / PostgreSQL only, while `closure_tree` supports these and sqlite in addition. So either would not help with regards to MS SQL server (which, to make things worse, apparently also has issues with the nested set as mentioned in the code and in #38184). I have no idea how large of an active SQL server user base Redmine has, but I guess just dropping support is probably not an option even if we scheduled this for 6.0.

Interestingly, from the closure_tree README:

> Database row-level locks work correctly with PostgreSQL, but MySQL's row-level locking is broken, and erroneously reports deadlocks where there are none. To work around this, and have a consistent implementation for both MySQL and PostgreSQL, with_advisory_lock is used automatically to ensure correctness.

So right now I am thinking about adding `with_advisory_lock` to place a proper global lock in the mysql-specific branch of `lock_nested_set` to solve the MySQL deadlock issues there.

I gave closure_tree a try, and I must say I like it because it stores the hierarchy in a separate table, and gets by with just deletes and inserts in that table, where our nested set has to update all following issues in a given set if one member is removed. It also was not too hard to plug into Redmine instead of issue_nested_set. I did not run into any consistency issues in my tests, but there were stale object errors caused by the `create_parent_issue_journal` hook when it bumps the `updated_on` on the parent issue (actually that is the same issue as with our current nested set). That alone is easy to fix, but then there's still the issue with lack of sqlserver support.

For now I attach two patches, one that adds with_advisory_lock to achieve the global lock instead of locking the settings table, and the second one which moves `create_parent_issue_journal` to an after_commit hook. This resolves any stale object errors, and just the consistency error when removing records from a nested set in parallel remains (if tx isolation == repeatable read).

Converting `handle_parent_change` in issue_nested_set.rb to an after_commit hook fixed the consistency issues but broke several other tests, mainly because other after_save hooks in the issue model rely on the hierarchy already being updated (i.e. when updating computed done ratios). Changing all that to also happen later, in after_commit, felt wrong to me. I am starting to think that the way the nested set is implemented right now is just not compatible with repeatable reads and we should either fix that by switching to a different storage model for the hierarchy (i.e., a closure tree with added sqlserver support?), or accept it as a (to be documented) limitation that can be fixed by changing the MySQL tx isolation.

#4 - 2023-11-06 21:17 - Marius BĂLTEANU

Jens Krämer wrote in #note-3:

> Hi Marius, thank you for looking into this!
>
> From what I can tell `typed_dag` supports MySQL / PostgreSQL only, while `closure_tree` supports these and sqlite in addition. So either would not help with regards to MS SQL server (which, to make things worse, apparently also has issues with the nested set as mentioned in the code and in #38184). I have no idea how large of an active SQL server user base Redmine has, but I guess just dropping support is probably not an option even if we scheduled this for 6.0.

Unfortunately, we don't have any analytics on this, but considering the fact that the tests are failing on MS SQL for some time (#39443), I suspect there may not be an active user base. Also, taking into account that we are only a fewer active contributors, maybe it is a good ideea to support only PostgreSQL, MySQL and maybe MariaDB along with SQLite for dev purposes.

#5 - 2023-11-06 21:31 - Marius BĂLTEANU

Until we get more feedback on the sqlserver topic and to have a quick win on this very old issue, maybe it's better to commit the patches and document the limitation along with the fix?

#6 - 2023-11-08 01:55 - Jens Krämer

Marius BALTEANU wrote in #note-5:

> Until we get more feedback on the sqlserver topic and to have a quick win on this very old issue, maybe it's better to commit the patches and document the limitation along with the fix?

Yes, I think that's a good idea.

#7 - 2023-11-08 05:39 - Mischa The Evil

- Related to Feature #35685: Support for MySQL > 5.7 or MariaDB added

#8 - 2023-11-12 12:12 - Marius BĂLTEANU

I would like to commit the fixes next week, but I'm not sure if we can deliver these fixes in minor releases (5.0.7 and 5.1.1) or we should do it only in a major/feature release. What do you think?

#9 - 2023-11-14 05:53 - Jens Krämer

Strictly speaking it's a bug fix, so adding it to a minor release would be fine in my book. I'd suggest to skip the test case added by me which still fails after this, together with a hint that this can be fixed by switching the tx isolation to 'read committed'.

#10 - 2023-11-17 08:27 - Marius BĂLTEANU

- Assignee set to Marius BĂLTEANU

#11 - 2023-11-18 23:47 - Marius BĂLTEANU

Both patches committed, including the test for concurrent subtask removal, I think it's importat to cover this case as well. In order to make to make the concurrency tests pass on MySQL, I've added an workaround to set the transaction_isolation to READ-COMMITTED only for those tests.

Working on this today, I saw some differences between the test results for concurrency tests without changing the isolation level to READ-COMMITTED:

- On MySQL 8 and 8.1, only the new test (@test_concurrent_subtask_removal) fails, the nested set became corrupted.
- On MySQL 5.7, all three tests fail.

Still in progress:

1. merge the fixes to stable branches
2. update Wiki page

#12 - 2023-11-19 08:13 - Go MAEDA

Marius BALTEANU wrote in #note-8:

> I would like to commit the fixes next week, but I'm not sure if we can deliver these fixes in minor releases (5.0.7 and 5.1.1) or we should do it only in a major/feature release. What do you think?

Since users of older versions of Redmine do not want big changes, I suggest that the change not be merged into 5.0-stable, but only into 5.1-stable.

Redmine 5.1.0 was just released at the end of last month and still has few users, so a quick release of 5.1.1 would effectively have the same effect as including this change only in the major release.

#13 - 2023-11-19 12:00 - Marius BĂLTEANU

- Target version changed from 6.0.0 to 5.1.1

#14 - 2023-11-19 12:05 - Marius BĂLTEANU

- Tracker changed from Patch to Defect

- Status changed from New to Resolved

Go MAEDA wrote in #note-12:

> Marius BALTEANU wrote in #note-8:
>
> > I would like to commit the fixes next week, but I'm not sure if we can deliver these fixes in minor releases (5.0.7 and 5.1.1) or we should do

> it only in a major/feature release. What do you think?

Since users of older versions of Redmine do not want big changes, I suggest that the change not be merged into 5.0-stable, but only into 5.1-stable.

Redmine 5.1.0 was just released at the end of last month and still has few users, so a quick release of 5.1.1 would effectively have the same effect as including this change only in the major release.

Merged only to 5.1-stable, thanks!

#15 - 2023-11-19 12:05 - Marius BĂLTEANU

- Resolution set to Fixed

#16 - 2023-11-20 23:38 - Marius BĂLTEANU

Jens Krämer wrote in #note-9:

> Strictly speaking it's a bug fix, so adding it to a minor release would be fine in my book. I'd suggest to skip the test case added by me which still fails after this, together with a hint that this can be fixed by switching the tx isolation to 'read committed'.

I've started a dedicated page (MySQL_configuration) in the Wiki to cover the isolation level change, please fell free to make changes to it. I will link this page to RedmineInstall.

I think we finished all the work required for this and we can close it.

#17 - 2023-11-20 23:39 - Marius BĂLTEANU

Jens Krämer, thanks again for you work on this issue, it is an important fix!

#18 - 2023-11-22 03:06 - Jens Krämer

I checked out the wiki page and I don't think there's anything to add. Thank you!

#19 - 2023-11-22 07:54 - Marius BĂLTEANU

- Status changed from Resolved to Closed

Jens Krämer wrote in #note-18:

> I checked out the wiki page and I don't think there's anything to add. Thank you!

Thanks! I will link the page before the release.

Files

0001-work-around-nested-set-related-deadlocks-in-MySQL-Ma.patch	4.02 KB	2023-10-30	Jens Krämer
0002-switch-tx-isolation-level-to-READ-COMMITTED-for-MySQ.patch	2.52 KB	2023-10-30	Jens Krämer
0001-use-a-global-lock-to-work-around-deadlock-errors-wit.patch	7.36 KB	2023-11-01	Jens Krämer
0002-moves-create_parent_issue_journal-to-after_commit.patch	2.97 KB	2023-11-01	Jens Krämer