Redmine - Defect #15947

Deadlock when delete issues in same time on multiple sessions

2014-01-23 11:48 - Nguyen Thinh Kha

Status:	Closed		**Start date:**	
Priority:	High		**Due date:**	
Assignee:	Jean-Philippe Lang		**% Done:**	0%
Category:	Issues		**Estimated time:**	0.00 hour
Target version:	2.5.0			
Resolution:	Fixed		**Affected version:**	2.3.3

Description

Dear team,

When we have multiple sessions try to delete issues in same times, we encounter this internal error:

> ActiveRecord::StatementInvalid (Mysql2::Error: Deadlock found when trying to get lock; try restarting transaction: SELECT `issues`.* FROM `issues` WHERE `issues`.`root_id` = 145786 AND (`issues`.`lft` >= 1 AND `issues`.`rgt` <= 2) AND (`issues`.id != 145786) ORDER BY `issues`.`lft`):

The trace:

> Started DELETE "/issues/17669" for 127.0.0.1 at 2014-01-23 16:34:09 +0700

Processing by IssuesController#destroy as HTML
Parameters: {"authenticity_token"=>"tlayCB7PIEzTrauenLr7g4HoeWkjz6S43YG6tZBRQE0=", "id"=>"17669"}
(3.0ms) SELECT MAX AS max_id FROM `settings`
User Load (3.0ms) SELECT `users`.* FROM `users` WHERE `users`.`type` IN ('User', 'AnonymousUser') AND `users`.`status` = 1 AND `users`.`id` = 1454 LIMIT 1
Current user: ttanthanh (id=1454)
Issue Load (3.0ms) SELECT `issues`.* FROM `issues` WHERE `issues`.`id` = 17669
Project Load (2.0ms) SELECT `projects`.* FROM `projects` WHERE `projects`.`id` = 10 LIMIT 1
EnabledModule Load (3.0ms) SELECT name FROM `enabled_modules` WHERE `enabled_modules`.`project_id` = 10
(2.0ms) SELECT SUM AS sum_id FROM `time_entries` WHERE (issue_id IN (17669))
Issue Load (4.0ms) SELECT `issues`.* FROM `issues` WHERE `issues`.`id` = 17669 LIMIT 1
(2.0ms) BEGIN
Journal Load (3.0ms) SELECT `journals`.* FROM `journals` WHERE `journals`.`journalized_id` = 17669 AND `journals`.`journalized_type` = 'Issue'
SQL (2.0ms) DELETE FROM `journal_details` WHERE `journal_details`.`journal_id` = 343792
SQL (3.0ms) DELETE FROM `journals` WHERE `journals`.`id` = 343792
SQL (2.0ms) DELETE FROM `time_entries` WHERE `time_entries`.`issue_id` = 17669
SQL (2.0ms) DELETE FROM `issue_relations` WHERE `issue_relations`.`issue_from_id` = 17669
SQL (3.0ms) DELETE FROM `issue_relations` WHERE `issue_relations`.`issue_to_id` = 17669
Issue Load (2.0ms) SELECT `lft`, `rgt`, `parent_id` FROM `issues` WHERE `issues`.`id` = 17669 LIMIT 1 FOR UPDATE
Issue Load (725.0ms) SELECT id FROM `issues` WHERE (`lft` >= 1) FOR UPDATE
Issue Load (3.0ms) SELECT `issues`.* FROM `issues` WHERE `issues`.`root_id` = 17669 AND (`issues`.`lft` >= 1 AND `issues`.`rgt` <= 2) AND (`issues`.id != 17669) ORDER BY `issues`.`lft`
SQL (3.0ms) UPDATE `issues` SET `lft` = (`lft` - 2) WHERE `issues`.`root_id` = 17669 AND (`lft` > 2) ORDER BY `issues`.`lft`
SQL (2.0ms) UPDATE `issues` SET `rgt` = (`rgt` - 2) WHERE `issues`.`root_id` = 17669 AND (`rgt` > 2) ORDER BY `issues`.`lft`
Issue Load (4.0ms) SELECT `issues`.* FROM `issues` WHERE `issues`.`id` = 17669 LIMIT 1
Attachment Load (3.0ms) SELECT `attachments`.* FROM `attachments` WHERE `attachments`.`container_id` = 17669 AND `attachments`.`container_type` = 'Issue' ORDER BY attachments.created_on ASC, attachments.id ASC
SQL (5.0ms) DELETE FROM `custom_values` WHERE `custom_values`.`customized_id` = 17669 AND `custom_values`.`customized_type` = 'Issue'
SQL (2.0ms) DELETE FROM `watchers` WHERE `watchers`.`watchable_id` = 17669 AND `watchers`.`watchable_type` = 'Issue'
Changeset Load (10.0ms) SELECT `changesets`.* FROM `changesets` INNER JOIN `changesets_issues` ON `changesets`.`id` = `changesets_issues`.`changeset_id` WHERE `changesets_issues`.`issue_id` = 17669 ORDER BY changesets.committed_on ASC, changesets.id ASC
SQL (3.0ms) DELETE FROM `issues` WHERE (`issues`.`id` = 17669 AND `issues`.`lock_version` = 1)
(9.0ms) COMMIT
Redirected to http://localhost:3001/projects/demoproject/issues

Completed 302 Found in 36813ms (ActiveRecord: 808.0ms)

We have two lines may be cause deadlock (I guess that)

    Issue Load (725.0ms) SELECT id FROM `issues` WHERE (`lft` >= 1) FOR UPDATE

This query lock table and cost too much time.

    Issue Load (3.0ms) SELECT `issues`.* FROM `issues` WHERE `issues`.`root_id` = 17669 AND (`issues`.`lft` >= 1 AND
    `issues`.`rgt` <= 2) AND (`issues`.id != 17669) ORDER BY `issues`.`lft`

And this query always return empty result.

Thanks for your support.

Associated revisions

Revision 12445 - 2013-12-22 16:49 - Jean-Philippe Lang

Fixed slow SELECT ... FOR UPDATE query when deleting an issue.

History

#1 - 2014-01-24 09:51 - Jean-Philippe Lang

- Status changed from New to Closed

- Assignee set to Jean-Philippe Lang

- Target version set to 2.5.0

- Resolution set to Fixed

This was fixed in r12445 for 2.5.0. Only the deleted issues are now locked:

```
SELECT id FROM `issues` WHERE `issues`.`root_id` = 13167 AND (`lft` >= 1) ORDER BY `issues`.`lft` FOR UPDATE
```

instead of:

```
SELECT id FROM `issues` WHERE (`lft` >= 1) FOR UPDATE
```