

Redmine - Defect #25273

Cannot create project/subproject

2017-03-07 00:00 - Carsten Weise

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Projects	Estimated time:	0.00 hour
Target version:		Affected version:	3.3.2
Resolution:	Invalid		

Description

When trying to create a project or subproject, redmine goes into an infinite loop (apparently - longest time I checked was 1 hour). Browser tab becomes inactive, saying "waiting for localhost". Memory consumption will increase, after a while it will slow down the browser and also the computer tremendously (probably due to swapping).

This will not only happen when using the browser GUI, but also when creating a project programmatically (from Ruby).

Last time I was able to add a project was in February. Only when I tried today, it did not work anymore.

My current configuration:

Environment:

Redmine version 3.3.2.stable

Ruby version 2.3.1-p112 (2016-04-26) [x86_64-linux]

Rails version 4.2.7.1

Environment production

Database adapter SQLite (3.17.0 2017-02-13)

SCM:

Subversion 1.9.5

Git 2.11.1

Filesystem

Redmine plugins:

no plugin installed

In fact, it happened to me while I still was on redmine 3.2, but I have upgraded now to 3.3.2, and the behavior is the same. Everything else works fine as far as I can see - I can add issues and delete projects, but not add new ones.

This happens with my current database. I have tried with a fresh database - I was able to create three projects, but then ran into the same problem.

My current guess is that this will be hard to reproduce. However, I would be nice to have some ideas where to start searching for the problem... for debugging I'm running from the command line, but I only get the usual hello messages, nothing else.

History

- #1 - 2017-03-07 15:54 - Carsten Weise
- Have been looking a little closer into the problem.
- The infinite loop happens when the project is stored using ActiveRecord::save. I can see that validate_parent is executed before the loop.
- So it seems to me the problem comes from the complex relations for Project.
- Not sure what is the best way to debug what is going wrong in the ActiveRecord method.
- #2 - 2017-03-08 21:52 - Jean-Philippe Lang
- Status changed from New to Needs feedback

This will not only happen when using the browser GUI, but also when creating a project programmatically (from Ruby).

Could you show the code you're using so we can try to reproduce?

#3 - 2017-03-14 15:46 - Carsten Weise

Sorry for replying late - been away for some days.

So I can reproduce the error with the following code (credentials for my login are of course the correct ones...)

```
require "rubygems"
require 'active_resource'
class Project < ActiveResource::Base
  puts "class Project"
  self.site = 'http://localhost:7777'
  self.user = 'theUser'
  self.password = 'thePassword'
  self.format = :xml
end
puts "retrieving all projects"
myProjects = Project.find(:all, :params => { :limit => 1000 })
puts "searching for project 'Test'"
testProject = nil
myProjects.each { |project|
  if (project.name == "Test") then
    testProject = project;
    break;
  end
}
if (testProject == nil) then
  puts "project 'Test' not found"
  exit -1
end
puts "Project #{testProject.name} has id #{testProject.id}"
puts "create new subproject for #{testProject.name}"
newProject = Project.new(
  :parent_id => testProject.id,
  :name => "Test Subproject",
  :identifier => "test_subproject"
)
puts "make new project persistent"
newProject.save
```

If I run this on my current redmine installation, I will get the following:

```
class Project
retrieving all projects
searching for project 'Test'
Project Test has id 17
create new subproject for Test
make new project persistent
```

This shows that I am connected to my redmine instance, and that retrieving projects is possible. After the last line, the script will not return, but wait for the save to return. When I go to the browser interface, it will tell me "waiting for localhost" as the server is busy with the save command.

The only way to make the system work again is killing the rails instance for redmine.

This will give me the following output:

```
/home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/protocol.rb:153:in `read_nonblock': end of file reached (
EOFError)
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/protocol.rb:153:in `rbuf_fill'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/protocol.rb:134:in `readuntil'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/protocol.rb:144:in `readline'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http/response.rb:39:in `read_status_line'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http/response.rb:28:in `read_new'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1414:in `block in transport_request'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1411:in `catch'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1411:in `transport_request'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1384:in `request'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1377:in `block in request'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:853:in `start'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1375:in `request'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1398:in `send_entity'
  from /home/zane/.rvm/rubies/ruby-2.2.3/lib/ruby/2.2.0/net/http.rb:1186:in `post'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/active_resource-4.0.0/lib/active_resource/connection.rb:121:in `b
```

```

lock in request'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/notifications.rb:164:in
`block in instrument'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/notifications/instrumen
ter.rb:20:in `instrument'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/notifications.rb:164:in
`instrument'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/connection.rb:118:in `r
equest'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/connection.rb:106:in `b
lock in post'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/connection.rb:216:in `w
ith_auth'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/connection.rb:106:in `p
ost'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/base.rb:1435:in `block
in create'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/callbacks.rb:88:in `__r
un_callbacks__'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/callbacks.rb:778:in `__r
un_create_callbacks'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/callbacks.rb:81:in `run
_callbacks'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/base.rb:1434:in `create
'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/observing.rb:21:in `cre
ate_with_notifications'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/base.rb:1224:in `block
in save'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/callbacks.rb:88:in `__r
un_callbacks__'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/callbacks.rb:778:in `__r
un_save_callbacks'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activessupport-4.2.7.1/lib/active_support/callbacks.rb:81:in `run
_callbacks'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/base.rb:1223:in `save'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/validations.rb:116:in `
save_with_validation'
  from /home/zane/.rvm/gems/ruby-2.2.3/gems/activeresource-4.0.0/lib/active_resource/observing.rb:21:in `sav
e_with_notifications'
  from makeProject.rb:32:in `<main>'

```

This way I can reproduce the problem on my installation, but I guess this will not work on a fresh database.

I saw postings that said that active resource can be a problem when you have complex relations between tables.

#4 - 2017-03-28 16:05 - Toshi MARUYAMA

- Status changed from Needs feedback to Closed
- Resolution set to Invalid

Do not use SQLite on production.

#5 - 2017-03-28 16:20 - Carsten Weise

- Status changed from Closed to Reopened

Could you be more specific? What is the problem with SQLite? I have other Rails apps running on my system with much larger databases, and they run fine.

#6 - 2017-03-28 16:41 - Toshi MARUYAMA

- Status changed from Reopened to Closed

Carsten Weise wrote:

Could you be more specific? What is the problem with SQLite? I have other Rails apps running on my system with much larger databases, and they run fine.

This is FAQ.

<https://www.google.com/search?q=why+not+use+sqlite+production>

#7 - 2017-03-28 18:13 - Carsten Weise

- Status changed from Closed to Reopened

Sorry, but I still do not understand what you try to say, and how it is related to my problem.

First, redmine FAQ does not say not to use sqlite - I cannot find that.

Second, the links I get when I use your google query basically say "sqlite might be a problem with high bandwidth and lots of parallel users, but it is perfectly suited for low bandwidth web servers".

I use redmine internally - low bandwidth and very few users. The problem occurs when there is only one user and only one pending request.

It's fine for me if there is a specific problem with redmine+sqlite, or rails+sqlite. However, so far I do not see this from your links.

#8 - 2017-03-28 19:58 - Toshi MARUYAMA

- Status changed from Reopened to Closed

Carsten Weise wrote:

First, redmine FAQ does not say not to use sqlite - I cannot find that.

[RedmineInstall](#):

SQLite 3 (not for multi-user production use!)

It's fine for me if there is a specific problem with redmine+sqlite, or rails+sqlite. However, so far I do not see this from your links.

<https://sqlite.org/faq.html#q5>

Multiple processes can have the same database open at the same time. Multiple processes can be doing a SELECT at the same time. But only one process can be making changes to the database at any moment in time

#9 - 2017-03-28 20:45 - Carsten Weise

- Status changed from Closed to Reopened

As I said before, the problem occurs with a single access. This is not multi-user production. So the RedmineInstall does not apply.

It's ok for me if you want to close this as not many are affected by the problem.

It's ok for me if you close this due to the fact that redmine does not support sqlite.

However, it's not ok for me to close this with an invalid reasoning.

I can only ask you to read the links you pointed to with your google query, e.g.

<http://charlesleifer.com/blog/five-reasons-you-should-use-sqlite-in-2016/>

"Don't believe the FUD about it not being production-worthy, or not being suitable for use in web-applications."

#10 - 2017-03-28 21:22 - Toshi MARUYAMA

- Status changed from Reopened to Closed

Carsten Weise wrote:

As I said before, the problem occurs with a single access. This is not multi-user production. So the RedmineInstall does not apply.

[#25273#note-3](#) says **multi-process** problem.

[source:test/unit/project_nested_set_test.rb](#) covers your situation.

Please ask Rails team whether Rails and sqlite gem support SQLite (3.17.0 2017-02-13).

I don't want to install such new SQLite and try to reproduce.