Redmine - Patch #23328
Improve Update/Create issue speed
2016-07-14 11:35 - Victor Campos

| Status: | New |
| Priority: | Normal |
| Assignee: | Jean-Philippe Lang |
| Category: | Performance |
| Target version: | Candidate for next major release |

Description

Hi guys,

When Redmine look for what members it should send e-mail, they interate one by one fetching principal. This is a N + 1 Query problem.

When we have more then 5K users in one project it is a problem. So with a single line change I drop the time for update issue from 5 to 2 seconds.

I hope this help you.

Date:   Tue Jul 12 19:37:14 2016 -0300
improve update/create speed
diff --git a/app/models/project.rb b/app/models/project.rb
index 660a486..88bd8eb 100644
--- a/app/models/project.rb
+++ b/app/models/project.rb
@@ -524,7 +524,7 @@
  
 # Returns the users that should be notified on project events
def notified_users
   # TODO: User part should be extracted to User#notify_about?
-   members.select {|m| m.principal.present? && (m.mail_notification? || m.principal.mail_notification == 'all')}.collect {|m|
   m.principal|
+   members.includes(:principal).select {|m| m.principal.present? && (m.mail_notification? || m.principal.mail_notification == 'all ')}).collect {|m| m.principal|
   end

   # Returns a scope of all custom fields enabled for project issues

History

#1 - 2016-07-14 19:53 - Lucas Arnaud
- File 0001-Improving-performance-of-project-notified_users-by-e.patch added

I resolved this issue a bit different. I changed the includes to eager_load to explicitly eager load the principal association and added a find_each to save memory when the quantity of members is to big.

```ruby
members.eager_load(:principal).find_each()
 .select {|m| m.principal.present? && (m.mail_notification? || m.principal.mail_notification == 'all')}
 .collect {|m| m.principal}
```

I've made some tests and these are the results:
<table>
<thead>
<tr>
<th># of project members</th>
<th>current method</th>
<th>after patch</th>
</tr>
</thead>
<tbody>
<tr>
<td>6024</td>
<td>6.13s</td>
<td>1.15s</td>
</tr>
<tr>
<td>7933</td>
<td>7.57s</td>
<td>1.40s</td>
</tr>
<tr>
<td>7935</td>
<td>7.46s</td>
<td>1.32s</td>
</tr>
</tbody>
</table>

#2 - 2016-07-15 15:31 - Victor Campos

Yes, for memory it’s a better solution.

=)

Thx for this patch

#3 - 2016-07-16 04:04 - Go MAEDA
- Description updated

#4 - 2016-07-16 04:21 - Go MAEDA
- Status changed from New to Needs feedback

Redmine 3.3.0 uses preload method in Project#notified_users. Please see r15518.
Could you test Redmine 3.3.0?

#5 - 2016-07-16 15:08 - Victor Campos

Go MAEDA wrote:

```
Redmine 3.3.0 uses preload method in Project#notified_users. Please see r15518.
Could you test Redmine 3.3.0?
```

Hi Go MAEDA,

What is the policy for update redmine stable branch? When 3.3-stable was lunch I update my redmine for it. When I read your comments I realided that there is a lot off new commits, with new features (redmine.lib changed a lot), performance issues fixed, etc.

About this issue, why preload and not eager_load? And I think the Lucas’s idea with find_each is good to prevent memory problems.

#6 - 2016-07-17 09:50 - Go MAEDA
- Status changed from Needs feedback to New
- Assignee set to Jean-Philippe Lang
- Target version set to Candidate for next major release

Thanks for the quick feedback.

Victor Campos wrote:

```
What is the policy for update redmine stable branch? When 3.3-stable was lunch I update my redmine for it. When I read your comments I realided that there is a lot off new commits, with new features (redmine.lib changed a lot), performance issues fixed, etc.
```
I am not a committer, so I can't explain about the policy. But as I know, the branch was used to prepare releasing of 3.3.0. Many revisions were merged from trunk before 3.3.0 is released.

About this issue, why preload and not eager_load? And I think the Lucas's idea with find_each is good to prevent memory problems.

I would like Jean-Philippe Lang to make a judgment. Setting assignee to Jean-Philippe.

### #7 - 2016-08-22 02:49 - Go MAEDA
- Category set to Performance

#### Files
- changeset_r619e156986dde1b674fa1e56bad4bc862c6e9df3.diff
  - 904 Bytes
  - 2016-07-14
  - Victor Campos
- 0001-Improving-performance-of-project-notified_users-by-e.patch
  - 1.1 KB
  - 2016-07-14
  - Lucas Arnaud