

Redmine - Patch #23334

Issue#editable_custom_field_values very slow for issues with many custom fields

2016-07-14 14:25 - Stephane Evr

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Performance	Estimated time:	0.00 hour
Target version:	3.3.1		

Description

I have several issues with around 50 custom fields. When doing updates to these issues, I identified poor performance on the `editable_custom_field_values` function being called to check if current user has access to the custom fields.

Here is the code for this method (in `app/model/issue.rb`):

```
# Returns the custom_field_values that can be edited by the given user
def editable_custom_field_values(user=nil)
  visible_custom_field_values(user).reject do |value|
    read_only_attribute_names(user).include?(value.custom_field_id.to_s)
  end
end
```

Here, it seems that `read_only_attribute_names(user)` is recomputed for each `visible_custom_field_values(user)` without caching, which will slow down the action if the issue has many custom fields associated (only if user is provided).

I have applied a simple patch (I store the result of `read_only_attribute_names(user)` before the loop):

```
def editable_custom_field_values(user=nil)
  read_only_attr_names_array = read_only_attribute_names(user)
  visible_custom_field_values(user).reject do |value|
    read_only_attr_names_array.include?(value.custom_field_id.to_s)
  end
end
```

Now, some results computed for an admin user, for approx 100 issues with various custom fields:

Before:

```
2.2.2 :001 > puts Benchmark.measure { u = User.find(88); Issue.last(100).to_a.map{|i| i.editable_custom_field_values(u)} }
2.2.2 :002 > 102.260000    0.710000 102.970000 (103.480136)
```

After:

```
2.2.2 :001 > puts Benchmark.measure { u = User.find(88); Issue.last(100).to_a.map{|i| i.editable_custom_field_values(u)} }
2.2.2 :002 > 5.070000    0.090000  5.160000 ( 5.198187)
```

Environment:

Redmine version	3.3.0.stable
Ruby version	2.2.2-p95 (2015-04-13) [x86_64-linux]
Rails version	4.2.6
Environment	development
Database adapter	Mysql2

Associated revisions

Revision 15742 - 2016-08-20 13:38 - Jean-Philippe Lang

Issue#editable_custom_field_values very slow for issues with many custom fields (#23334).

Patch by Stephane Evr.

Revision 15774 - 2016-08-31 19:00 - Jean-Philippe Lang

Merged r15742 (#23334).

History

#1 - 2016-07-15 14:16 - Victor Campos

Good, I will try this.

#2 - 2016-07-15 18:41 - Victor Campos

Victor Campos wrote:

Good, I will try this.

It works here, thx =)

#3 - 2016-07-16 04:00 - Go MAEDA

- Target version set to 3.3.1

Looks good to me. Caching a value is reasonable. Setting target version to 3.3.1.

Here is a diff made from Stephane Evr's post.

Index: app/models/issue.rb

```
=====
--- app/models/issue.rb      (revision 15663)
+++ app/models/issue.rb      (working copy)
@@ -574,8 +574,9 @@
 
     # Returns the custom_field_values that can be edited by the given user
     def editable_custom_field_values(user=nil)
+   read_only_attr_names_array = read_only_attribute_names(user)
       visible_custom_field_values(user).reject do |value|
-       read_only_attribute_names(user).include?(value.custom_field_id.to_s)
+       read_only_attr_names_array.include?(value.custom_field_id.to_s)
     end
   end
```

#4 - 2016-08-12 05:16 - Toshi MARUYAMA

- Description updated

#5 - 2016-08-20 13:38 - Jean-Philippe Lang

- Subject changed from *Issues editable_custom_field_values method very slow for issues with many custom fields* to *Issue#editable_custom_field_values very slow for issues with many custom fields*

- Status changed from *New* to *Resolved*

- Assignee set to *Jean-Philippe Lang*

Change committed in [r15742](#), thanks.

#6 - 2016-08-31 19:00 - Jean-Philippe Lang

- Status changed from *Resolved* to *Closed*