Redmine - Defect #19924

Adding subtask takes very long

2015-05-22 15:46 - Sebastian Paluch

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Jean-Philippe Lang	**% Done:**	0%
Category:	Performance	**Estimated time:**	0.00 hour
Target version:	3.3.0		
Resolution:	Fixed	**Affected version:**	3.0.1

Description

One of our project have a parent task to group 1192 subtasks...

Showing the tasks details takes about 21s, most of it is in issues/show.html.erb.

```
Started GET "/redmine/issues/4821" for 127.0.0.1 at 2015-05-22 14:51:31 +0200
Processing by IssuesController#show as HTML
  Parameters: {"id"=>"4821"}
  Current user: XXX (id=10)
  Rendered issues/_action_menu.html.erb (0.0ms)
  Rendered issues/_action_menu.html.erb (0.0ms)
  Rendered attachments/_form.html.erb (0.0ms)
  Rendered issues/_edit.html.erb (0.0ms)
  Rendered plugins/redmine_agile/app/views/agile_boards/_issues_sidebar.html.erb (0.0ms)
  Rendered plugins/redmine_agile/app/views/agile_charts/_agile_charts.html.erb (0.0ms)
  Rendered issues/_sidebar.html.erb (0.0ms)
  Rendered issues/show.html.erb within layouts/base (20747.9ms)
  Rendered plugins/sidebar_hide/app/views/sidebar/_hideButton_partial.html.erb (0.0ms)
Completed 200 OK in 21341ms (Views: 19312.7ms | ActiveRecord: 1747.2ms)
```

Adding new subtasks takes even more than 80s, most of it in ActiveRecord. This is real problem.

```
Started GET "/redmine/projects/XXX/issues?query_id=1617" for 127.0.0.1 at 2015-05-22 14:46:10 +020
0
Processing by IssuesController#index as HTML
  Parameters: {"query_id"=>"1617", "project_id"=>"XXX"}
  Current user: XXX (id=154)
  Rendered queries/_filters.html.erb (93.6ms)
  Rendered queries/_columns.html.erb (15.6ms)
  Rendered mailer/_issue.text.erb (0.0ms)
  Rendered mailer/issue_add.text.erb within layouts/mailer (0.0ms)
  Rendered mailer/_issue.html.erb (15.6ms)
  Rendered mailer/issue_add.html.erb within layouts/mailer (15.6ms)
Redirected to http://XXX/redmine/issues/22455
Completed 302 Found in 84240ms (ActiveRecord: 75192.0ms)
```

The group that is using it claims this was working fine in Redmine 2.x.

Now, we are running Bitnami 3.0.1 installation:

```
Environment:
  Redmine version                3.0.1.stable
  Ruby version                   2.0.0-p594 (2014-10-27) [i386-mingw32]
  Rails version                  4.2.0
  Environment                    production
  Database adapter               Mysql2
SCM:
  Subversion                     1.8.10
  Git                            1.9.5
  Filesystem
```

```
Redmine plugins:
  inside_avatar                 1.0.2
  parent_issue_filter           1.0.1
  redmine_agile                 1.3.8
  redmine_spent_time_column     2.1.0
  redmine_version_fixed_issues  1.0.0
  sidebar_hide                  0.0.7
```

History

#1 - 2015-06-18 20:52 - Sebastian Paluch

After some research, it seems that the reason why it takes so long is The Nested Set Model used to build subtasks hierarchy.

Adding every single subtask needs an update of *lft* and *rgt* fields of every task in the database located on the right and up in the tree!!! and currently we have over 20000 issues.

The improvement could be, as suggested by Mike, "to look at using index (*lft* and *rgt*) numbers that increment by 1,000 instead of by one, when you need to add a node you place it between two existing index numbers, then occasionally during slow periods you re-index everything."

I do not know how many issues average Redmine installation has and if subtasks are commonly used but if performance starts to degrade at 20000 issues this is worth to improve it.

#2 - 2015-06-18 21:56 - Jean-Philippe Lang

- Status changed from New to Needs feedback

The nested set model for Redmine issues is scoped under each root. So adding a subtask updates the issues that are under the same root only, not all issues. 1192 subtasks under a same parent is not a common situation, I'll try to reproduce but I'd really like to see the whole log in debug mode to see what is happening.

#3 - 2015-06-19 11:20 - Sebastian Paluch

- File production.zip added

Right, I did forget about the scoping under the root issue.
I have attached the log showing 3 operations: show the issue details, show new subtask form and POST request. It is captured on different machine so the timings may be a bit different.
I could not find a single SQL request that takes such long time. It seems that it is a sum of huge number of individual requests.

#4 - 2015-07-03 17:01 - Sebastian Paluch

Can I provide anything more to help debugging it?

#5 - 2016-06-08 20:36 - Sebastian Paluch

Jean-Philippe,
is there any chance to take a look on this?

#6 - 2016-06-08 21:56 - Jean-Philippe Lang

Sorry, I'll work on this. Can you tell how the 1192 subtasks are organized under the parent task (a flat list or multiple levels of grouping)? And how many members in this project?

In the issue details, I can see **a lot** of queries generated by some plugin (agile_colors).

Sebastian Paluch wrote:

> After some research, it seems that the reason why it takes so long is The Nested Set Model used to build subtasks hierarchy.

I can see in your log that updating the nested set is not the issue, it takes 15.6ms. The query also shows that only the subtasks are updated, not all the issues present in the database (WHERE `issues`.`root_id` = 4821):

```
SQL (15.6ms)[0m  UPDATE `issues` SET lft = CASE WHEN lft >= 4396 THEN lft + 2 ELSE lft END,
rgt = CASE WHEN rgt >= 4396 THEN rgt + 2 ELSE rgt END WHERE `issues`.`root_id` = 4821 AND (lft >= 4396 OR rgt
>= 4396)
```

#7 - 2016-06-09 15:47 - Sebastian Paluch

- File production-no-plugins-issue-show-new-show.zip added

It seems that since I was testing it the number of subtasks has grown, there is now 3696 issues orginized in a tree with 4 levels of depth where most

of them are on 1 & 2. There is 87 project members.

```
mysql> select count(1) from issues where root_id=4821;
+----------+
| count(1) |
+----------+
|     3696 |
+----------+
1 row in set (0.00 sec)

mysql> select count(1) as count, (SELECT (COUNT(parents.id) - 1) FROM issues AS parents WHERE issues.lft BETWE
EN parents.lft AND parents.rgt AND issues.root_id
= parents.root_id) as depth from issues where issues.root_id=4821 group by depth;
+-------+-------+
| count | depth |
+-------+-------+
|     1 |     0 |
|  1837 |     1 |
|  1558 |     2 |
|   300 |     3 |
+-------+-------+
4 rows in set (13.74 sec)

mysql> select count(1) from members inner join projects on projects.id=members.project_id inner join issues on
 issues.project_id=projects.id where issues.id=482
1;
+----------+
| count(1) |
+----------+
|       87 |
+----------+
1 row in set (0.00 sec)
```

The table *agile_colors* is from RedmineCRM Agile plugin. It could have some impact on "show" but it should not have impact on "new". The "new" is most painful as it takes minutes.

I have removed all plugins and run it, this is how long adding subtask took (full log attached):

```
Completed 302 Found in 93976ms (ActiveRecord: 85036.7ms)
```

#8 - 2016-06-12 14:11 - Jean-Philippe Lang

- Status changed from Needs feedback to Closed

- Assignee set to Jean-Philippe Lang

- Target version set to 3.3.0

- Resolution set to Fixed

Thanks for the feedback. I was able to reproduce this behaviour with Redmine 3.0 and Redmine 3.2 as well, this is due to thousands of queries on issues and issue_relations tables. The good news is that this problem is fixed in 3.3 (by this commit: r15056). Here are the results on my machine with an issue that has 3200 subtasks on 3 levels:

Redmine 3.0 / 3.2 (this could be dramatically slower depending on the database):

```
Completed 302 Found in 12210ms (ActiveRecord: 4093.2ms)
```

Redmine 3.3 with the same data:

```
Completed 302 Found in 353ms (ActiveRecord: 61.0ms)
```

Showing the issue details with all its subtasks is still a bit slow (2 ou 3 seconds) but we can't make the rendering of thousands of links much faster. We should probably limit the number of subtasks that are displayed on the issue details and add a link to the issue list with a filter on parent issue to show a paginated list of the subtasks.

I'm closing it as it seems to be fixed, please reopen if you're still experiencing problems after upgrading.

Files

production.zip	51.2 KB	2015-06-19	Sebastian Paluch
production-no-plugins-issue-show-new-show.zip	63.3 KB	2016-06-09	Sebastian Paluch