

Redmine - Defect #34957

Search issues by custom fields is slow

2021-03-25 18:58 - Alejandro Grijalba

<b>Status:</b>	New	<b>Start date:</b>											
<b>Priority:</b>	Normal	<b>Due date:</b>											
<b>Assignee:</b>		<b>% Done:</b>	0%										
<b>Category:</b>	Performance	<b>Estimated time:</b>	0.00 hour										
<b>Target version:</b>		<b>Affected version:</b>											
<b>Resolution:</b>													
<b>Description</b>													
<p>We have more than 30k issues, some of them use custom fields.</p> <p>We found that searching by custom fields is slow, takes about 20 seconds the search.</p> <p>It is slow both searching using API, or searching using GUI.</p> <p>The custom field in this case is of type integer.</p> <p>The problem seems to be related to the SQL query that is being done during search.</p> <p>The query looks like this:</p> <pre>Select * from issues, ...   AND issues.id IN (SELECT issues.id FROM    issues LEFT OUTER JOIN custom_values                   ...                   )</pre> <p>The code for the subquery is at models/issue_custom_field.rb</p> <p>There seem to be performance problems with "IN" subqueries in MySQL (<a href="https://stackoverflow.com/a/9627668/366209">https://stackoverflow.com/a/9627668/366209</a>).</p> <p>It looks like the subquery cannot be cached in MySQL, only the whole query can. (And that doesn't even help, because SQL cache is flushed when a ticket is created anyway)</p> <p>Ironically the suquery alone runs in less than a second, and the external query alone is fast too.</p> <p>So maybe the suquery could be reworked to use JOIN instead of IN, that's what they suggest in Stack Overflow. That worked in my tests, when I ran modified SQL queries manually.</p> <p>I wish I could change the code, but it seems a little complicated for me right now.</p> <p>Our Redmine was installed from a Bitnami's virtual machine template.</p>													
<b>Environment:</b>													
<table><tr><td>Redmine version</td><td>3.3.2.stable</td></tr><tr><td>Ruby version</td><td>2.1.10-p492 (2016-04-01) [x86_64-linux]</td></tr><tr><td>Rails version</td><td>4.2.7.1</td></tr><tr><td>Environment</td><td>production</td></tr><tr><td>Database adapter</td><td>Mysql2</td></tr></table>				Redmine version	3.3.2.stable	Ruby version	2.1.10-p492 (2016-04-01) [x86_64-linux]	Rails version	4.2.7.1	Environment	production	Database adapter	Mysql2
Redmine version	3.3.2.stable												
Ruby version	2.1.10-p492 (2016-04-01) [x86_64-linux]												
Rails version	4.2.7.1												
Environment	production												
Database adapter	Mysql2												
<b>MySQL Version:</b> 5.6.35													

History

#1 - 2021-03-28 08:12 - Go MAEDA

Alejandro Grijalba wrote:

So maybe the suquery could be reworked to use JOIN instead of IN, that's what they suggest in Stack Overflow. That worked in my tests, when I ran modified SQL queries manually.  
I wish I could change the code, but it seems a little complicated for me right now.

Thank you for report the issue. Could you post the SQL you modified?