Redmine - Patch #26122
Implementation of visible conditions with inner join instead of subselect
2017-06-07 22:51 - Pavel Rosický

Status: Resolved
Priority: Normal
Assignee:
Category: Performance
Target version: Candidate for next major release

Start date:
Due date:
% Done: 0%
Estimated time: 0.00 hour

Description

The change from #21608 should be reverted because it's a speed regression:

1/ current version

   EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em. NAME = 'issue_tracking')

2/ previous version (actually faster even without an index on enabled modules)

   projects.id IN (SELECT project_id FROM enabled_modules em WHERE em.project_id = projects.id AND em. NAME = 'issue_tracking')

3/ fastest version (in some cases better indexes were used)

   INNER JOIN `enabled_modules` ON `enabled_modules`.project_id = `projects`.id WHERE `enabled_modules`.name = 'issue_tracking'
   or

   INNER JOIN `enabled_modules` ON `enabled_modules`.project_id = `projects`.id AND `enabled_modules`.name = 'issue_tracking'

Patches:
- revert of #21608, because the previous version was faster (about 30%)
- new index "enabled_modules_name", but it's not very helpful unless you have many projects (3000+)
- implementation of visible conditions with inner join instead of subselect, it passes all tests, but it should be refactored. I want to know what do you think about it first

Rails version  4.2.8
Ruby version  2.1.9-p490 (x64-mingw32)
RubyGems version  2.6.12
Rack version  1.6.8
Middleware  Rack::Sendfile, Rack::ContentLength, ActionDispatch::Static, Rack::Lock,
 #=>ActiveSupport::Cache::Strategy::LocalCache::Middleware::0x00000007745610>, Rack::Runtime, Rack::MethodOverride,
ActionDispatch::RequestId, Rails::Rack::Logger, ActionDispatch::ShowExceptions, ActionDispatch::DebugExceptions,
ActionDispatch::RemoteIp, ActionDispatch::Reloadable, ActionDispatch::Callbacks,
ActiveRecord::ConnectionAdapters::ConnectionManagement, ActiveRecord::QueryCache, ActionDispatch::Cookies,
ActionDispatch::Session::CookieStore, ActionDispatch::Flash, ActionDispatch::ParamsParser,
ActionDispatch::XmlParamsParser, Rack::Head, Rack::ConditionalGet, Rack::ETag, RequestStore::Middleware,
OpenIdAuthentication
Environment  development
Database adapter  mysql2
Database schema version  20170607051650
I did some additional benchmarking and I was wrong. Exists really performs better.

so the only relevant patch is joins_enabled_modules.patch

Issue.visible is faster on my environment with this patch about 10% compared to exists with all caching disabled. I don't think it's worth to use it on other places.

Edited issue properties based on the feedback and slightly improved the markup of the description.

As a speed regression, I suggest to put this on version#130

Well, it heavily depends on what filters and other conditions are used, it isn't easy to measure. For my long time observations using INNER JOINS is slightly faster (especially on big tables like issues) and mysql has better working plans with it.

On the other side, joins can't be passed directly into WHERE statement which is what allowed_to_condition should do - it increases code complexity. I want some opinions if you think it's worth to use it or not.

@Fernando - as discussed before, it's not a speed regression just an improvement, so 3.5.0 is fine.

Target version changed from 4.1.0 to Candidate for next major release

Pavel Rosický wrote:

Well, it heavily depends on what filters and other conditions are used, it isn't easy to measure. For my long time observations using INNER JOINS is slightly faster (especially on big tables like issues) and mysql has better working plans with it.

On the other side, joins can't be passed directly into WHERE statement which is what allowed_to_condition should do - it increases code complexity.
complexity. I want some opinions if you think it's worth to use it or not.

@Fernando - as discussed before, it's not a speed regression just an improvement, so 3.5.0 is fine.

I'm removing this from version:"4.1.0" because it is not a simple change and without performance tests that can be run before and after on both mysql and postgres, it is hard to say if the code complexity added by the patch it's worth it or not. Please correct me if I am wrong.

#9 - 2019-09-27 18:22 - Pavel Rosicky
- Status changed from New to Resolved

no, I agree. Thanks for the feedback, this can be closed.

Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>20170607051650_add_index_on_enabled_modules_name.rb.patch</td>
<td>416 Bytes</td>
<td>2017-06-07</td>
<td>Pavel Rosicky</td>
</tr>
<tr>
<td>revert_exists.patch</td>
<td>959 Bytes</td>
<td>2017-06-07</td>
<td>Pavel Rosicky</td>
</tr>
<tr>
<td>joins_enabled_modules.patch</td>
<td>11 KB</td>
<td>2017-06-07</td>
<td>Pavel Rosicky</td>
</tr>
</tbody>
</table>