

ステータス:	New	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Performance_53	予定工数:	0.00時間
対象バージョン:		作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/24473	status_id:	1
category_id:	53	tracker_id:	1
version_id:	0	plus1:	0
issue_org_id:	24473	affected_version:	
author_id:	123866	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	5		

説明			
Environment:			
Redmine version		3.3.0.stable	
Ruby version		2.2.1-p85	(2015-02-26) [x86_64-linux]
Rails version		4.2.6	
Environment		production	
Database adapter		Mysql2	

So I tried to investigate why, when exporting an issue with lots of custom fields, my log file would look like this:

CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]
CACHE (0.0ms)	SELECT	`issues`.*	FROM	`issues`	WHERE	`issues`.`id` = 94261	LIMIT 1	[[{"id", 94261}]]

There is one line per custom field apparently.

My investigation led this method (in lib/redmine/field_format.rb):

```
def formatted_custom_value(view, custom_value, html=false)
  formatted_value(view, custom_value.custom_field, custom_value.value, custom_value.customized, html)
```

end

When custom_value.customized is called, this leads to the Customized object being loaded. Here, Customized is an issue, and it would be reloaded once per custom field.

Some caching would be needed here, where we directly provide the customized object to this method, although I am not sure what's best to do...

journals

I have attached a (dirty) patch which fixed the problem for me. The export time is basically twice faster.

related_issues

relates,New,24477,CSV export feature for thousand of issues

relates,New,31076,Issues CSV / PDF export via ActiveJob

duplicates,Closed,24755,Slow Queries because of Custom Fields

履歴

#1 - 2022/05/10 17:06 - Admin Redmine

- カテゴリ を Performance_53 にセット