redmineorg-copy202205 - Vote #75575

Searches should be twice faster

2022/05/09 17:24 - Admin Redmine

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:	Search engine_16	**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/17889	**status_id:**	1
category_id:	16	**tracker_id:**	2
version_id:	0	**plus1:**	0
issue_org_id:	17889	**affected_version:**	
author_id:	80618	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	
comments:	3		

説明

When we run a search, we can see in the debug log that for each category to look into (issues, forums, wiki...) there are 2 queries which are almost the same, and take the same time. One is for counting results, the other one is for getting the first 11 lines.
This is discussed in http://www.redmine.org/boards/1/topics/41475 for example.

In /lib/plugins/acts_as_searchable.rb, line 126 and 128, we have (I think) the origin of those 2 queries:

l 126: results_count = scope.count
l 128: scope_with_limit = scope.limit(options[:limit])

Wouldn't it be possible to include a column like

"total=COUNT(*) OVER()"

(MSSQL or Oracle syntax) to get that results_count in only one query? I tested the query, and it takes the same time with or without the total column, even if on a large table (7000ms of query execution).

I mean, instead of having 2 queries:

[1m[35m (7805.9ms)[0m EXEC sp_executesql N'SELECT COUNT(DISTINCT [issues].[id]) FROM ...))))))

and

[1m[36mSQL (7592.5ms)[0m [1mEXEC sp_executesql N'SELECT TOP (11) [issues].id FROM ...)))))) GROUP BY [issues].id, issues.id ORDER BY issues.id DESC'[0m

We would have only one query:

[1m[36mSQL (7592.5ms)[0m [1mEXEC sp_executesql N'SELECT TOP (11) [issues].id, total=COUNT(*) OVER() FROM ...)))))) GROUP BY [issues].id, issues.id ORDER BY issues.id DESC'[0m

And that would make searches twice faster.

journals

@OVER()@ is not supported in Sqlite3 and MySQL, so this is probably hard to implement in the general case. It might however be feasible as an optimization for MS SQL and PotgreSQL with different code paths respectively.

In MySQL you can use @FOUND_ROWS()@: http://dev.mysql.com/doc/refman/5.7/en/information-functions.html#function_found-row

S
related_issues relates,Closed,18631,Better search results pagination

履歴

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

- カテゴリ を Search engine_16 にセット