ChiliProject - Bug # 1069: Better query optimization

Status:	Closed	**Priority:**	Low
Author:	David Kowis	**Category:**	
Created:	2012-07-08	**Assignee:**	
Updated:	2012-10-23	**Due date:**	
Remote issue URL:			
Affected version:			

Description:

I'm not exactly sure whether this should be a bug or a feature request, and I'm not completely sure where the problem is either, heh.

I'm looking at our chiliproject installation at www.sourcemage.org that is operating on a reasonably large git repo, Very large actually.

<pre>
mysql> show processlist;
+-----+-------------+-----------+-------------+---------+------+---------------+---+
| Id | User | Host | db | Command | Time | State | Info |
+-----+-------------+-----------+-------------+---------+------+---------------+---+
162	chiliproject	localhost	chiliproject	Query	20	Sorting result	SELECT * FROM `changesets` WHERE (id < 82009 AND repository_id = 7) ORDER BY id DESC LIMIT 1
172	chiliproject	localhost	chiliproject	Sleep	1		NULL
176	root	localhost	chiliproject	Query	0	NULL	show processlist
177	chiliproject	localhost	chiliproject	Query	12	Sorting result	SELECT * FROM `changesets` WHERE (id < 62784 AND repository_id = 7) ORDER BY id DESC LIMIT 1
178	chiliproject	localhost	chiliproject	Sleep	2		NULL
180	chiliproject	localhost	chiliproject	Query	32	Sorting result	SELECT * FROM `changesets` WHERE (id > 66988 AND repository_id = 7) ORDER BY id ASC LIMIT 1
+-----+-------------+-----------+-------------+---------+------+---------------+---+
6 rows in set (0.00 sec)

mysql> explain SELECT * FROM `changesets` WHERE (id > 66988 AND repository_id = 7) ORDER BY id ASC LIMIT 1;
+----+-------------+------------+------+---+-------------+---------+-------+-------+----------------------------+
| id | select_type | table | type | possible_keys | key | key_len | ref | rows | Extra |
+----+-------------+------------+------+---+-------------+---------+-------+-------+----------------------------+
| 1 | SIMPLE | changesets | ref | PRIMARY,changesets_repos_rev,index_changesets_on_repository_id,changesets_repos_scmid | changesets_repos_rev | 4 | const | 40239 | Using where; Using filesort |
+----+-------------+------------+------+---+-------------+---------+-------+-------+----------------------------+
1 row in set (0.02 sec)
</pre>

As you can see, there's some pretty darn big queries going on there. Unfortunately the query also isn't able to use the indexes as indicated by the "Using filesort" part. Looking through the MySQL documentation, it's because the sql query is touching a field that's not being ordered by.

I don't know if there's really anything that can be done here to optimize this some, but it'd certainly improve speed on our system :)

Any advice, or where I could hack in a different query would be good.

Thanks,
David

History

2012-07-08 11:10 pm - Felix Schäfer

Do you use the auto-update for the repository view (default) or do you have a cron/hook updating the changesets info in ChiliProject for you?

One way or the other, your installation seems to have a problem with file permissions anyway: looking at "one of the repository views":http://www.sourcemage.org/projects/sorcery/repository, it's missing the commit information and all the info for the directories, which most of the time is an indication that the ChiliProject instance can't access the git repos.

Last thing: I don't think we have any MySQL-savvy devs around, if you know someone who could help us that one (should we order by both columns touched in the WHERE, does it lack a proper index for that, and so on). It might be a query generated by rails though, so there's a chance we can't change it much.

2012-07-09 01:33 am - David Kowis

It's auto-updated by default.

We don't have permissions problems, I've explicitly disabled that additional information, because it would take minutes for those pages to load making all the calls out to git.

It might be a query generated by rails, but I think you can instead explicitly use a query instead of relying on Arel (if you're even at Arel stuff yet? Probably not I think chiliproject is rails 2.x) to grab the data.

2012-07-09 10:09 am - Felix Schäfer

David Kowis wrote:
> It's auto-updated by default.

OK, so try disabling that (global settings > repositories > automatically fetch changesets) and see if the view is still taking so long. If that reduces the time the view takes to generate (it should visibly reduce it), you should consider updating your repo information outside of the page visits. There's an [[FAQ#How-come-I-have-to-visit-the-repository-module-before-I-see-any-changes|FAQ entry]] that should shed some light (and somehow I thought I had written a longer explanation and how to somewhere, but I can't seem to find itâ€¦), the gist of it being: fetching the changesets is an expensive operation and is coupled to the repository view so that it works "out of the box", on bigger installs it should be decoupled and triggered either via cron or post-commit hooks. The local variant is:

<pre><code>rake redmine:fetch_changesets RAILS_ENV=production
</code></pre>

This will need to spin up a new rails process and you can't limit it to a specific repo/project, the other possibility is via a web hook:

<pre><code>curl "${CHILI_URI}/sys/fetch_changesets?key=${CHILI_API_KEY}&id=${PROJECT_IDENTIFIER}"
</code></pre>

The later method has the advantage that it reuses a web process, so no need to spin up a new rails process and all, and that you can direct it to a specific project/repository (you can also not specify an @id=@, which then updates all repos), the disadvantage being that in certain circumstances it might block all your web processes.

The 2 methods I've seen to use those were either through cron (update the repository information every so often) or through a post-commit (svn)/post-receive (git) hook. I use the web hook variant in a post-receive hook, that allows for incremental and only-when-needed updates to ChiliProject, it works pretty well (you might want to start the curl command in a subprocess @curl foo &@ though so as not to have the potentially expensive operation block the git push).

I hope this was clear enough, if not feel free to ask for clarification :-)

> We don't have permissions problems, I've explicitly disabled that additional information, because it would take minutes for those pages to load making all the calls out to git.

Ugh, you should have told us then, meddling with the internals, especially for performance reasons, is a good indication that there's more work for us to be done ;-)

> It might be a query generated by rails, but I think you can instead explicitly use a query instead of relying on Arel (if you're even at Arel stuff yet? Probably not I think chiliproject is rails 2.x) to grab the data.

No ARel yet indeed, and we could use an explicit query either way. I'd like to understand the problem first though and hear from someone with more MySQL knowledge how to best deal with this.

2012-07-09 02:33 pm - David Kowis

Felix SchÃ¤fer wrote:
> David Kowis wrote:
> I hope this was clear enough, if not feel free to ask for clarification :-)

Yeah, doing that will help with part of it, but it's not directly related to the querying issue. :)

>
> > We don't have permissions problems, I've explicitly disabled that additional information, because it would take minutes for those pages to load making all the calls out to git.
>
> Ugh, you should have told us then, meddling with the internals, especially for performance reasons, is a good indication that there's more work for us to be done ;-)

Well, you guys already know about it: https://github.com/chiliproject/chiliproject/blob/stable/lib/redmine/scm/adapters/git_adapter.rb#L22 I just made that false so that it wouldn't report that stuff :)

Also: https://www.chiliproject.org/issues/317 Once upon a time I thought I would have time to work on this, but the investigation I did pointed me at libgit2, which has native ruby bindings, rather than shelling it out to git directly, or using something broken like grit.

>
> > It might be a query generated by rails, but I think you can instead explicitly use a query instead of relying on Arel (if you're even at Arel stuff yet? Probably not I think chiliproject is rails 2.x) to grab the data.
>
> No ARel yet indeed, and we could use an explicit query either way. I'd like to understand the problem first though and hear from someone with more MySQL knowledge how to best deal with this.

Well, the problem in this particular case is that the query cannot use the indexes, and so it has to sort 45,000 rows each time, just to get one row, because it's using fields in the where clause that aren't in the order by. It's entirely possible that if you also order by the other field in the where clause, it'll be able to use the indexes. I haven't tried explaining that query.

2012-10-23 03:51 pm - David Kowis

- Status changed from Open to Closed

This turned out to be an artifact of old ruby, and old mysql driver. Updating to a newer driver, and everything sucks a whole lot less. I'm going to close.