

Redmine - Defect #1435

slow display of 'repository' tab for huge repository (pagination problem?)

2008-06-12 17:12 - Thomas Capricelli

Status:	Closed	Start date:	2008-06-12
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:	SCM	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Fixed		
Description			
i've added a (mercurial) repository with some thousands of revision and each time (not only the first one) i go on the repository tab, it takes ~80 seconds before being displayed. No modifications were made between those tests.			
You can clone the repository using this line if you want to reproduce/test: hg clone http://sources.freehackers.org/hg.cgi/Yzis			
(and i can be reached as 'orzel' on irc)			
Related issues:			
Related to Redmine - Defect #1931: Timeout risks: need asynchronous operations		New	2008-09-22
Related to Redmine - Defect #3449: Redmine Takes Too Long On Large Mercurial ...		Closed	2009-06-05
Related to Redmine - Defect #7047: Git adapter very slow when a commit modifi...		New	2010-12-04
Related to Redmine - Defect #6013: git tab,browsing, very slow -- even after ...		Closed	2010-08-02
Related to Redmine - Defect #8857: Git: Too long in fetching repositories aft...		Closed	2011-07-20

Associated revisions

Revision 2043 - 2008-11-17 18:27 - Jean-Philippe Lang

Do not query multiple times git for branch (#1435).

History

#1 - 2008-06-14 13:05 - Jean-Philippe Lang

I've cloned your repository. The first access takes a few minutes (changesets are loaded in the db). But after that, displaying the repository is almost immediate. Tested with trunk and 0.7 stable branch. Which version/revision are you using ? Also, please attach the full log (in debug mode) generated when you click the repository tab.

#2 - 2008-08-04 20:48 - Jean-Philippe Lang

- Status changed from New to Closed

No feedback. Reopen if needed.

#3 - 2008-10-03 01:39 - Thomas Capricelli

- % Done changed from 0 to 100

Today i've updated (to [r1923](#)), and it works a lot better. I probably had a too old version. Keep up the good work.

#4 - 2008-10-03 05:01 - David Kowis

I can say this does happen with a large git repo.

<http://shlrn.org/redmine/repositories/show/grimoire-test-1>

That's running latest svn ([r1923](#)) using passenger on an apache server.

It takes a ridiculously long time to do the initial import.

one can clone the repo I'm using there:
git clone git://shlrn.org/omfggrimoire.git

Further accesses take tens of seconds to render the page. Granted, this is a huge repo, but it does happen.

#5 - 2008-11-17 13:22 - Burt Culver

- Status changed from Closed to Reopened

This happens for me too on a Git repository. I just added it to our redmine today, loaded the change sets on the command line (6900 of them). Now whenever I click on Repository the site hangs and the browser times out. In debug mode I see that it takes redmine 248 seconds to complete - each and every time.

The delay seems to come from redmine doing a "git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' log --date=iso --pretty=fuller -1 master --faq.php" call on all 300 top level files that the "git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' ls-tree -l 'HEAD:'" lists.

I upgraded to the latest in trunk ([r2042](#)) to see if there was a fix but it didn't help.

I'm pretty sure it is the number of files in the directory that make the difference rather than the revision count as David Kowis's example has what looks to be 100 files in his top-level directory and it is about 1/3 as fast. Redmine's own svn repo trunk/app/views with 37 folders takes a few seconds to load.

Here is a summary of the debug output for our repo:

```
Processing RepositoriesController#show (for 127.0.0.12 at 2008-11-18 01:01:26) [GET]
  Session ID: a6af16ec88ef0b45e2e5c6a53188149a
  Parameters: {"action"=>"show", "id"=>"bgz", "controller"=>"repositories"}
  SQL (0.000101) SELECT max(`settings`.updated_on) AS max_updated_on FROM `settings`
  User Load (0.000057) SELECT * FROM `users` WHERE (`users`.`id` = 3) AND (status = 1)
  Project Load (0.000061) SELECT * FROM `projects` WHERE (`projects`.`identifier` = 'bgz') LIMIT 1
  Repository Load (0.000045) SELECT * FROM `repositories` WHERE (`repositories`.project_id = 1) LIMIT 1
  EnabledModule Load (0.000065) SELECT * FROM `enabled_modules` WHERE (`enabled_modules`.project_id = 1)
  Shelling out: git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' log --raw --date=iso --pretty=fuller -n 1
  Changeset Load (0.000130) SELECT * FROM `changesets` WHERE (`changesets`.repository_id = 4) ORDER BY changesets.committed_on DESC, changesets.id DESC LIMIT 1
  Changeset Load (0.000062) SELECT * FROM `changesets` WHERE (`changesets`.`scmid` = '268b9f207d6dbb880f353b1e1a934e926140dc5e') AND (`changesets`.repository_id = 4) ORDER BY changesets.committed_on DESC, changesets.id DESC LIMIT 1
  Shelling out: git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' ls-tree -l 'HEAD:'
  Shelling out: git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' branch
  Shelling out: git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' log --date=iso --pretty=fuller -1 master -- '.cvsignore'
  Shelling out: git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' branch
  Shelling out: git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' log --date=iso --pretty=fuller -1 master -- '.gitignore'
  --the branch and log commands repeat for the 300 files--
  Changeset Load (0.000159) SELECT * FROM `changesets` WHERE (`changesets`.repository_id = 4) ORDER BY committed_on DESC, changesets.committed_on DESC, changesets.id DESC LIMIT 10
  Rendering template within layouts/base
  Rendering repositories/show
  Changeset Load (0.000123) SELECT * FROM `changesets` WHERE (`changesets`.`revision` = 'f5f46f9f01c29a3414260e83fa483d75d45813bd') AND (`changesets`.repository_id = 4) ORDER BY changesets.committed_on DESC, changesets.id DESC LIMIT 1
  User Load (0.000059) SELECT * FROM `users` WHERE (`users`.`id` = 9)
  Changeset Load (0.000082) SELECT * FROM `changesets` WHERE (`changesets`.`revision` = '59dd7ab40a541ace8256ff5f1af9d50d39b1aed6') AND (`changesets`.repository_id = 4) ORDER BY changesets.committed_on DESC, changesets.id DESC LIMIT 1
  User Load (0.000058) SELECT * FROM `users` WHERE (`users`.`id` = 51)
  Changeset Load (0.000104) SELECT * FROM `changesets` WHERE (`changesets`.`revision` = '0c8ba9ce1711e9887786ec9b9d2b017659a24ab3') AND (`changesets`.repository_id = 4) ORDER BY changesets.committed_on DESC, changesets.id DESC LIMIT 1
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 51)
  --sql like this occurs 300 or so times--
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 4)
  Rendered repositories/_dir_list_content (2.26420)
  Rendered repositories/_dir_list (2.26477)
  UserPreference Load (0.000118) SELECT * FROM `user_preferences` WHERE (`user_preferences`.user_id = 3) LIMIT 1
  IT 1
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 10)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 4)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 4)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 4)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 42)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 42)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 42)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 42)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 4)
  CACHE (0.000000) SELECT * FROM `users` WHERE (`users`.`id` = 42)
```

```

CACHE (0.000000)  SELECT * FROM `users` WHERE (`users`.`id` = 10)
Rendered repositories/_revisions (0.04287)
Token Load (0.000123)  SELECT * FROM `tokens` WHERE (`tokens`.user_id = 3 AND (action='feeds')) LIMIT 1
SQL (0.000117)  SELECT count(DISTINCT `members`.id) AS count_all FROM `members` LEFT OUTER JOIN `projects`
ON `projects`.id = `members`.project_id LEFT OUTER JOIN `roles` ON `roles`.id = `members`.role_id WHERE (`memb
ers`.user_id = 3 AND (projects.status=1))
Project Load Including Associations (0.000110)  SELECT `projects`.`id` AS t0_r0, `projects`.`name` AS t0_r1
, `projects`.`description` AS t0_r2, `projects`.`homepage` AS t0_r3, `projects`.`is_public` AS t0_r4, `project
s`.`parent_id` AS t0_r5, `projects`.`projects_count` AS t0_r6, `projects`.`created_on` AS t0_r7, `projects`.`u
pdated_on` AS t0_r8, `projects`.`identifier` AS t0_r9, `projects`.`status` AS t0_r10, `parents_projects`.`id`
AS t1_r0, `parents_projects`.`name` AS t1_r1, `parents_projects`.`description` AS t1_r2, `parents_projects`.`h
omepage` AS t1_r3, `parents_projects`.`is_public` AS t1_r4, `parents_projects`.`parent_id` AS t1_r5, `parents_
projects`.`projects_count` AS t1_r6, `parents_projects`.`created_on` AS t1_r7, `parents_projects`.`updated_on`
AS t1_r8, `parents_projects`.`identifier` AS t1_r9, `parents_projects`.`status` AS t1_r10 FROM `projects` LEF
T OUTER JOIN `projects` parents_projects ON `parents_projects`.id = `projects`.parent_id INNER JOIN members ON
projects.id = members.project_id WHERE ((`members`.user_id = 3) AND ((projects.status=1)))
Rendered layouts/_project_selector (0.00469)
SQL (0.000106)  SELECT count(*) AS count_all FROM `versions` WHERE (`versions`.project_id = 1)
Wiki Load (0.000048)  SELECT * FROM `wikis` WHERE (`wikis`.project_id = 1) LIMIT 1
SQL (0.000042)  SELECT count(*) AS count_all FROM `boards` WHERE (`boards`.project_id = 1)
Completed in 224.48472 (0 reqs/sec) | Rendering: 2.71917 (1%) | DB: 0.01270 (0%) | 200 OK [http://redmine.fish
pond.co.nz/repositories/show/bgz]

```

#6 - 2008-11-17 18:41 - Jean-Philippe Lang

The delay seems to come from redmine doing a "git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' log --date=iso --pretty=fuller +
~~master~~ - 'faq.php'" call on all 300 top level files that the "git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' ls-tree -l 'HEAD:'" lists.

Indeed. I've just committed an improvement in [r2043](#) that prevents Redmine from running the same git branch command multiple times.
 Response time is almost divided by 2. I know that it doesn't solve your problem (120s would be still very slow), but it can help.
 Displaying a git directory with 300 files (with a small number of commits) dropped from 9s to 5s on my dev box.

Redmine's own svn repo trunk/app/views with 37 folders takes a few seconds to load.

Actually, it takes about 0.5s to generate the page for trunk/app/views (see X-Runtime server response header).
 But svn works in a different way. There's no need to query each entry to retrieve its last revision (a single command can be used to list the directory
 and get these revisions for each entry).

Do you know if git offers this possibility ?

#7 - 2008-11-17 22:41 - Burt Culver

Thanks for that update. I tried hardcoding the branch last night, but it only brought the response time down to 200 seconds. The main issue is that
 each git log on our server takes 1 to 2 seconds to complete. Per the GIT doco this is because it is looking at all the revisions ever and then just
 returning the last one for the file specified. I couldn't find a way to list a directory of the latest revisions. Mind you, we just switched from cvs to git so I
 am no expert on git at this point. The only way I could think of to speed this up would be to modify the database and at the time you load the revisions
 you query the revision and store what files were modified by each revision. There didn't seem to be an easy way to do this and would probably
 involve parsing the output of git show on the revision. Then, given the file name you could look up the latest revision in the database. Alternatively,
 you could do the git logs in parrallel rather than serially but even that would bog down the system.

#8 - 2008-11-17 23:13 - Jean-Philippe Lang

The only way I could think of to speed this up would be to modify the database and at the time you load the revisions you query the revision and
 store what files were modified by each revision.

File changes for each commit are already stored in the database

Then, given the file name you could look up the latest revision in the database.

I've just made a quick test, querying the database using the file path is slower on my machine that using git log (git log doesn't take more than 0.1s
 with a 30k commit git repo). Maybe it could be improved by adding some indexes on the table, I'll do some further tests.
 How many commits have you in your repository ?

#9 - 2008-11-17 23:27 - Burt Culver

We only have 7k changesets and 15k changes.

But git log takes considerably longer on my machine:

```
time git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' log --date=iso --pretty=fuller 1-master 'wikibase.png'
commit 59dd7ab40a541ace8256ff5f1af9d50d39b1aed6
Author: fishpond <fishpond>
AuthorDate: 2005-11-10 20:59:59 +0000
Commit: fishpond <fishpond>
CommitDate: 2005-11-10 20:59:59 +0000
```

Initial revision

```
real 0m2.042s
user 0m0.000s
sys 0m0.000s
bash-3.1$
```

#10 - 2008-11-18 00:00 - Burt Culver

git logs on newer files return much faster than the old dog I reported above:

```
time git --git-dir '/mnt/data-store/redmine/git-repositories/fishpond/' log --date=iso --pretty=fuller 1-master stylesheet.css.php
commit c2ffc2367e15b91e834e5dcdfacba6db8d8812cd
Author: James K. McFall <xxx>
AuthorDate: 2008-11-11 14:01:48 +1300
Commit: James K. McFall <xxx>
CommitDate: 2008-11-11 14:01:48 +1300
```

```
Dev 2296 - Improvements to product details page
Reviewed by Kelvin
```

```
real 0m0.020s
user 0m0.000s
sys 0m0.000s
```

or perhaps its the difference between imported from cvs commits and commits since we switched to git.

#11 - 2008-11-18 10:29 - Thomas Lecavelier

Added: relation to [#1931](#). It could hide performance problems.

#12 - 2008-11-18 23:26 - Burt Culver

I sent an email to git-list to get the expert opinion on how to handle it. Here is a link to the thread: <http://marc.info/?l=git&m=122698291625045&w=2>

Basically, git operates totally differently on the backend so doing the browsing like redmine does is not well supported at this point. The existing git tree browsers seem to be very commit rather than file based. For example: <http://git.kernel.org/> - once you go into a project, you start out with the latest commits and can click on tree to get to a file view (I think the file view at the time of that commit). Then when you click on a file's history you can see the commits. This is a radically different view of a tree than what redmine supports because git tracks changes in a radically different manner than the traditional version control systems (cvs, svn, etc).

So, I don't really know how to go forward. If you can send me the patch you used to do a db lookup for the entities rather than the system call - I think that would be faster on our system. Otherwise, maybe redmine could optionally skip displaying the latest commit on the repository browse for git repos? That would be a quick fix for this.

#13 - 2008-11-20 20:00 - Jean-Philippe Lang

- File *git_log_from_db.patch* added

Thanks for the info.

Here is the patch. It's not thoroughly tested but it seems to work.

First access may be slow (could be improved by adding an index on the "path" column of the table "changes"). But once there is some cache at db level, it's pretty fast.

Note that it doesn't give the exact same results as using git log on the file (eg. a git commit do not report to change a file when running git log <commit> but some files are reported to be modified by the commit when running git log -- <path>). Maybe there's a reason for this, but I'm really not mastering git.

#14 - 2008-11-24 08:17 - Burt Culver

Added that patch and the index. This is the loadtime now:

```
Processing RepositoriesController#show (for 127.0.0.12 at 2008-11-24 20:12:55) [GET]
Session ID: 67198a021a844af2bde909f20ec96b3d
Parameters: {"action"=>"show", "id"=>"bgz", "controller"=>"repositories"}
Rendering template within layouts/base
Rendering repositories/show
Completed in 2.78889 (0 reqs/sec) | Rendering: 2.21187 (79%) | DB: 0.12612 (4%) | 200 OK [http://redmine.fishpond.co.nz/repositories/show/bgz]
```

A giant improvement. Thanks Jean-Philippe.

#15 - 2009-12-03 21:53 - Thomas Capricelli

According to the last comment, it can be closed again... right ? Anyway, as the original author of the bug, I consider it solved.

Though it seems it went beyond my original problem..... so i let you decide.

#16 - 2010-03-12 15:54 - Toshi MARUYAMA

The reason of this Mercurial problem is "**Mercurial tip is not latest revision**".

I described at <http://www.redmine.org/issues/4455#note-23>.

I fixed this problem at [#4455](#).

#17 - 2010-10-04 21:56 - Marc Mengel

I've applied this patch on my server, as well as adding and index on `create(path)`; without even adding the index, it was a 3x speedup, and the `create index create_path` on `create(path)` got me another factor of 2.

Is this going to go into future releases?

#18 - 2010-10-05 00:50 - Toshi MARUYAMA

This issue contains several problems.

Mercurial

- fetching revisions performance problem ([#3449](#), [#3567](#))

It is resolved at <http://www.redmine.org/issues/4455#note-144>

Git

- fetching revisions performance problem ([#6013](#))
- browsing directory performance problem ([#5096](#))

#19 - 2010-10-05 01:25 - Toshi MARUYAMA

JPL's patch seems to fix git browsing directory performance problem ([#5096](#)).

For committing this patch, I think we need to make unit and functional tests and test repository.

And for adding index, I think we need to make db migration such as

source:tags/1.0.2/db/migrate/20100221100219_add_index_on_changesets_scmid.rb

#20 - 2010-10-06 18:12 - Marc Mengel

- *File `git_log_from_db.patch` added*

Okay, with this slight variation on the patch (attached) (which relies on `changeset_id`'s being monotonically increasing) and

`create index changes_path_id on changes(path,changeset_id)`

I get really quite snappy performance.

#21 - 2010-10-06 18:22 - Marc Mengel

Actually, belay that last one... ordering by `changeset_id` does not consistently get you the most recent change, in fact in many cases it seems to get you the **oldest** one...

#22 - 2010-10-06 22:29 - Marc Mengel

This time Rocky, for sure!

#23 - 2010-10-06 22:31 - Marc Mengel

- *File `better_git_log_from_db.patch` added*

#24 - 2010-10-06 22:33 - Marc Mengel

- *File `really_better_git_log_from_db.patch` added*

#25 - 2010-10-07 00:34 - Toshi MARUYAMA

Redmine 1.0 git adapter support git branch, so we need to consider it.

For example, <http://github.com/marutosi/redmine/blob/246a913ba3252977ff1eb813d793a3c0bb1323ba/app/models/repository/mercurial.rb#L69>

#26 - 2010-10-07 01:43 - Toshi MARUYAMA

Unit tests

- [source:tags/1.0.2/test/unit/repository_git_test.rb](#)
- [source:tags/1.0.2/test/unit/lib/redmine/scm/adapters/git_adapter_test.rb](#)

Functional test

- [source:tags/1.0.2/test/functional/repositories_git_controller_test.rb](#)

Test repository

- [source:tags/1.0.2/test/fixtures/repositories/git_repository.tar.gz](#)

#27 - 2011-01-06 16:02 - Toshi MARUYAMA

- *Status changed from Reopened to Closed*
- *Resolution set to Fixed*

Original Mercurial fetching revisions performance problem is fixed by [r4611](#).
I close this issue.

Git issues are followings.

- git fetching revisions performance problem ([#6013](#))
- git browsing directory performance problem ([#7047](#))

Files

git_log_from_db.patch	1.9 KB	2008-11-20	Jean-Philippe Lang
git_log_from_db.patch	1.88 KB	2010-10-06	Marc Mengel
better_git_log_from_db.patch	2.35 KB	2010-10-06	Marc Mengel
really_better_git_log_from_db.patch	2.33 KB	2010-10-06	Marc Mengel