

Convert the git adapter to rugged

2022/05/09 15:03 - Admin Redmine

ステータス:	New	開始日:	2010/10/04
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	SCM_3	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/6566	status_id:	1
category_id:	3	tracker_id:	2
version_id:	0	plus1:	0
issue_org_id:	6566	affected_version:	
author_id:	3866	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	15		

説明

There's been some talk about converting the git adapter to use git, but nothing concrete yet. I'll give it a try myself and post the link to the git branch here.

There have also been concerns about the speed of grit, but seeing it is used by github and implements stuff either as shell-outs or direct access depending on speed, I'd say it is fast enough for us. I will nonetheless try to provide a performance test to see how both adapters perform against a real-world git repository (probably against redmine.git ;-).

I've added the people whom I think might be interested in this matter as watchers, feel free to unwatch if this doesn't apply to you.

journals

When we added git branch support, there were a few patches that used grit. Might be useful to dig through that code and see if there is anything we can use.

I was rather planning on making a "clean" adapter, but I'll sure have a look :-)

This would be interesting if you can pull it off, although the few minutes I spent poking at Grit (and redmine.git) recently doesn't fill me with confidence. It went something like this...

```
$ irb
ruby-1.8.7-p249 > require 'grit'
=> true

ruby-1.8.7-p249 > repo_path = '/home/michael/rails/redmine'
=> "/home/michael/rails/redmine"

ruby-1.8.7-p249 > grit = Grit::Repo.new(repo_path)
=> #

ruby-1.8.7-p249 > recent_list = grit.commits_since(start = 'master', since = '2010-10-10', extra_options = {})
SystemStackError: stack level too deep
## full traceback omitted ...

ruby-1.8.7-p249 > recent_list_alt = `cd #{repo_path}; git log --since='2010-10-10' --format='%H'`.split
## succeeded. full array of results omitted ...

ruby-1.8.7-p249 > recent_list_alt.count
=> 81

ruby-1.8.7-p249 > Grit.version
=> "2.3.0"
```

Digging a little deeper revealed that Grit attempts to load every commit in the repo, and then handle your constraints (since = DATE, etc.) by filtering the complete list in ruby before returning it to you. Apparently redmine.git is a little too large for that approach to work (not even taking performance into account).

Hopefully I'm just doing it wrong and you'll have better luck with it. :)

Removing it from the Unplanned features since it doesn't seem to be usable and is no longer updated (last release is 2 years old).

Not saying it works, but last release was just yesterday (2.4.1)

<https://github.com/mojombo/grit/> so it certainly is being actively developed.

Right. I was looking at rubyforge :/

Assigned issue with no assignee back to New status.

Please change this issue to "Covert the git adapter to rugged". Rugged is the oficial ruby library with ruby bindings for libgit2. You can find more about it here:

<https://github.com/libgit2/rugged>

I haven't found a way to make searches by date without having to walk over all commits, but found some useful ways of walking over the commit trees (to get for example from x to y hashes, that is probably what's going to be used to get latest commits:

```
1.9.3-p327 > require 'rugged'
=> true

1.9.3-p327 > repo_path = '/path/to/redmine'
=> "/path/to/redmine"

1.9.3-p327 > repo = Rugged::Repository.new(repo_path)
=> #

1.9.3-p327 > walker = Rugged::Walker.new(repo)
=> #

1.9.3-p327 > walker.push("301d7e7cb13a09854a7c87aecedfcbfd27eb98d3")
1.9.3-p327 > walker.hide("c31f498ba6a21fd3e5ce7b9ba2f3b3cdc1b2e05b")

1.9.3-p327 > result = walker.to_a
=> [#, #, #]
```

or you can just iterate over for a more memory friendly operation:

```
1.9.3-p327 > walker.push("301d7e7cb13a09854a7c87aecedfcbfd27eb98d3")
1.9.3-p327 > walker.hide("c31f498ba6a21fd3e5ce7b9ba2f3b3cdc1b2e05b")
```

```
1.9.3-p327 > walker.each {|c| puts c.inspect}
#
#
#
```

A commit object have the following methods:

```
=> [:message, :time, :committer, :author, :tree, :parents, :to_hash, ..., :oid, :type, ...]
```

```
1.9.3-p327 > c.to_hash
=> {:message=>"German translation updated by Daniel Felix (#10191)/n/ngit-svn-id: svn+ssh://rubyforge.org/var/svn/redmine/trunk@10998 e93f8b46-1217-0410-a6f0-8f06a7374b81 /n", :committer=>{:name=>"Toshi MARUYAMA", :email=>"marutosijp2@yahoo.co.jp", :time=>2012-12-14 14:47:46 UTC}, :author=>{:name=>"Toshi MARUYAMA", :email=>"marutosijp2@yahoo.co.jp", :time=>2012-12-14 14:47:46 UTC}}
```

e=>2012-12-14 14:47:46 UTC}, :tree=>#, :parents=>[#]}

1.9.3-p327 > c.oid
=> "301d7e7cb13a09854a7c87aecedfcbfd27eb98d3"

1.9.3-p327 > c.type
=> "commit"

1.9.3-p327 > Rugged::Version
=> "0.16.0"

Gabriel Mazetto wrote:

Please change this issue to "Covert the git adapter to rugged".

Done.

Typo

I decided to try making the conversion... and have some questions:

Do I have to use "AbstractAdapter"? It was designed to be used with command line in mind, which is not the case with rugged. Should I try to create a lean "AbstractAdapter" with libraries in mind? I saw another issue talking about libsvn for example, that could be the potential for this new abstract library, with only the "SCM actions" in mind.

<https://github.com/brodock/redmine/commit/457322722a1e73ec3f13eec12afc311dc6b2f810>

Branches and Tags

(tests OK)

related_issues

relates,Closed,7146,Git adapter lost commits before 7 days from database latest changeset

relates,Closed,6013,git tab,browsing, very slow -- even after first time

relates,New,5357,Git: SCM revisions ordered by date/time (should be reverse commit order)

履歴

#1 - 2022/05/10 17:21 - Admin Redmine

- カテゴリ を SCM_3 にセット