Redmine - Feature #3389
Increase speed of Repository.fetch_changesets

2009-05-19 18:36 - Gregor Bader

Status:	Closed	**Start date:**	2009-05-19
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	SCM	**Estimated time:**	0.00 hour
Target version:	0.9.0		
Resolution:	Fixed		

Description

We are running Redmine trunk version r2743 with a local Subversion server on a Win2K virtual machine. The mongrel server and SQLite is used. I have setup a post-commit hook to run

```
ruby script/runner "Repository.fetch_changesets" -e production
```

to automatically update after each commit.
The problem is, that this call runs for about 20s - 30s and blocks the whole commit!!

Is there a way to speed things up? It only have to fetch 1 changeset (the one of current commit).
Perhaps another interface where I directly provide the log message could solve this?
Any other suggestions?

Related issues:

Related to Redmine - Feature # 2925: Url to fetch changesets for a repository	**Closed**	**2009-03-07**

History

#1 - 2009-05-19 20:57 - Eric Davis

I would suggest removing that from your post commit hook. Or backgrounding the process:

```
(ruby script/runner "Repository.fetch_changesets" -e production) &
```

#2 - 2009-05-19 23:17 - Bob Roberts

Eric Davis wrote:

> *I would suggest removing that from your post commit hook. Or backgrounding the process:*
>
> *[...]*

I agree, though not ideal, it's very annoying to block commits for 10s+. We use a windows "scheduled task" to run this every hour or so, and plan to change this to run 1 minute after the last commit (so that many commits at the same time don't cause a whole mass of ruby processes to start!). We also have a post-propchange commit hook to update revisions which is directly run from the hook but this is so rare it's not a big user issue that it takes 10-20 secs to run.

#3 - 2009-05-20 07:59 - Gregor Bader

Eric Davis wrote:

> *I would suggest removing that from your post commit hook. Or backgrounding the process:*

> *[...]*

We thought of backgrounding, too. But under windows this is not a trivial task! I tried several ways (start batchfile) and so on which create a new process, but subversion somehow tracks child processes and only finishes commit after all child processes are done.

But we found some program which works (CPAU http://www.joeware.net/freetools/tools/cpau/index.htm).

I have investigated the process and found out that not the svn fetch causes this delay, it is the ruby startup of the script!
Perhaps it is possible to provide a script which doesn't require much gems, so the startup would be faster?

#4 - 2009-06-04 15:12 - Bruno Medeiros

Gregor Bader wrote:

> *We thought of backgrounding, too. But under windows this is not a trivial task! I tried several ways (start batchfile) and so on which create a new process, but subversion somehow tracks child processes and only finishes commit after all child processes are done.*

Yes, i really don`t know how to schedule some script on windows, but should be a way..
I have Redmine set up on a Fedora 10 system, and have the following lines in my /etc/crontab to schedule my script:

```
*/2 9-19 * * * root /usr/bin/ruby /usr/redmine/script/runner "Repository.fetch_changesets" -e production
*/5 6-8 * * * root /usr/bin/ruby /usr/redmine/script/runner "Repository.fetch_changesets" -e production
*/5 20-23 * * * root /usr/bin/ruby /usr/redmine/script/runner "Repository.fetch_changesets" -e production
```

From 9am to 7pm we have more people working, so we decrease the time between runs to 2 minutes. 6am to 9am and 8pm to 0am we set the time between runs to 5 minutes. Other times we don`t run the script.

Each run takes about 12 seconds to execute here (a single core Atlhon64 3000+ CPU). It seems to be not much optimized for me, maybe some soft code would be better and allows to decrease more the time between runs (Sometimes people complaint about the delay between the commit and the refresh on tickets).

#5 - 2009-11-28 16:55 - Nikolay Kotlyarov

The same problem:(

We have several projects with several repositories. And not all users have permissions to access all that repositories.
So fetch-hook fails when such a user tries to commit, because it fetches ALL repositories (and it is superfluous work):

- fetching must use "project" parameter, because there is only one repository for every certain project.

And perhaps it would be faster when using existing ruby process(fetching repository by clicking on "Repository" tab in browser **is** fast)

#6 - 2009-11-28 17:49 - Jean-Philippe Lang
- Status changed from New to Closed
- Resolution set to Fixed

See r3107. You can now do a simple GET on an URL to fetch the changesets of all or one project.
"Enable WS for repository management" setting must be set to true.

It should be must faster than script/runner since it uses existing ruby process.

#7 - 2009-11-30 18:53 - Nikolay Kotlyarov

Jean-Philippe Lang wrote:

> *See r3107.*

Thank you very much:)

#8 - 2009-12-09 02:38 - Eric Davis

Jean-Philippe Lang wrote:

> *It should be must faster than script/runner since it uses existing ruby process.*

One note, this will tie up a web process while it's working. So if you are running under webrick (single process), the site will not load until the repository processing completes.

#9 - 2009-12-30 18:04 - Eric Davis

- Target version set to 0.9.0

#10 - 2010-04-12 21:56 - Bruno Medeiros

Sorry for the dumb questio, but I didn't find any docs of how to use it.
I'm trying to do a 'http://redmine.mycompany.com.br/sys/fetch_changesets?key=DoUMvOSpkUwDZQCnNCBw' but getting a 403. What's the correct way to do the request?

Thanks!

#11 - 2010-04-12 23:04 - Bruno Medeiros

Sorry, I did generate a new key and saved the configs again and now it seems to be working.

#12 - 2010-04-12 23:43 - Felix Schäfer

Bruno Medeiros wrote:

> *Sorry for the dumb questio, but I didn't find any docs of how to use it.*
> *I'm trying to do a 'http://redmine.mycompany.com.br/sys/fetch_changesets?key=DoUMvOSpkUwDZQCnNCBw' but getting a 403. What's the correct way to do the request?*
>
> *Thanks!*

I think you have to enable the *Repository management WS*, and not sure you need the key for it (but it doesn't hurt to have it). IIRC the /sys/ actions are not password- or key-protected and you need to take care of disallowing access to it on your own (in apache config restrict it to only one source ip for example).

#13 - 2010-04-28 23:34 - Bruno Medeiros

Just to help other people, here is my crontab file:

```
# Rodando script do Redmine que faz o fetch dos commits do SVN
*/1 9-19 * * * root wget http://redmine.company.com.br/sys/fetch_changesets?key=hfFhgfUytfjVjGfFJJGF -O - > /dev/null
*/2 6-8 * * * root wget http://redmine.company.com.br/sys/fetch_changesets?key=hfFhgfUytfjVjGfFJJGF -O - > /dev/null
*/2 20-23 * * * root wget http://redmine.company.com.br/sys/fetch_changesets?key=hfFhgfUytfjVjGfFJJGF -O - > /dev/null
*/30 0-5 * * * root wget http://redmine.company.com.br/sys/fetch_changesets?key=hfFhgfUytfjVjGfFJJGF -O - > /dev/null
```

#14 - 2012-04-18 13:40 - Andrew Lansdowne

Is there any locking done by this web service to ensure that fetch_changesets doesn't run concurrently?

We have a post commit hook on svn set up to call fetch_changesets to update the revisions for all projects, which works fine. But we just experienced an issue where we added a new repository to a project, and redmine stopped responding. Then svn commits hanged because they was waiting on redmine web service to return.

I am thinking that multiple people committed while redmine was updating its repository data, and each call to fetch_changesets would start fetching the same changesets while the previous one was still processing?

Is that a possibility? I have disabled our post commit hook to avoid this for the time being. I may implement locking within the hook itself so it doesn't call the web service concurrently. Thanks