Redmine - Feature #4482

Cache textile rendering

2009-12-24 18:39 - Eric Davis

Status:	Closed	**Start date:**	2009-12-24
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Text formatting	**Estimated time:**	0.00 hour
Target version:	1.0.0 (RC)		
Resolution:	Fixed		

Description

To improve performance, the core Textile rendering should be cached into HTML and stored in the database. We will still need to run Redmine's syntax parser in order to get Redmine specific markup (e.g. #1, r100) but that can be on the fly.

A big concern is to make sure the Redmine markup isn't cached, in order to preserve private data. (e.g. a link to a issue on a private project isn't shown to an unauthorized user).

Associated revisions

Revision 3253 - 2009-12-27 11:52 - Jean-Philippe Lang

Moves attachments parsing after textile parsing so that:

- attachments parsing does not rely on textile syntax
- textile output can be cached (#4482)

History

#1 - 2009-12-28 10:11 - Jean-Philippe Lang

- Target version deleted (1.0.0 (RC))

For most pages, Textile rendering is not an issue.

For example, **turning off** textile makes the rendering of average tickets only 10~15% faster depending of their text size. If you add the AR overhead to retrieve the cached HTML from the database, there will only be a small improvement.
Of course, this would be a bit more useful on big wiki pages but the database is definitely not the most effective way to cache textile rendering.

#2 - 2009-12-30 00:23 - Eric Davis

Jean-Philippe Lang wrote:

> For most pages, Textile rendering is not an issue.

I don't know if I agree with that. The statistics I'm getting are showing about 70% of the time being spent in the page rendering. We might need to do some profiling to make sure.

> but the database is definitely not the most effective way to cache textile rendering.

Where would you propose caching to? I could think of four places but they would be deployment specific (except the database):

1. to disk - needs a writable filesystem
2. to memcached - needs a memcache server setup
3. to memory - could cause the memory size of Redmine to grow as the cache grows
4. to database - would increase the size of the database and the data returned from queries

#3 - 2010-01-09 19:58 - Jean-Philippe Lang

Eric Davis wrote:

> > For most pages, Textile rendering is not an issue.
>
> I don't know if I agree with that. The statistics I'm getting are showing about 70% of the time being spent in the page rendering. We might need to do some profiling to make sure.

It doesn't mean that 70% of the time is spent in Textile. Rendering views is slow.

> but the database is definitely not the most effective way to cache textile rendering.

Where would you propose caching to? I could think of four places but they would be deployment specific (except the database):

1. to disk - needs a writable filesystem
2. to memcached - needs a memcache server setup
3. to memory - could cause the memory size of Redmine to grow as the cache grows
4. to database - would increase the size of the database and the data returned from queries

We can simply use Rails.cache. It uses MemoryStore by default but users can configure it to use whatever they want.

#4 - 2010-01-15 18:05 - Eric Davis

Jean-Philippe Lang wrote:

> It doesn't mean that 70% of the time is spent in Textile. Rendering views is slow.

Correct, that's why some profiling will be needed.

> We can simply use Rails.cache. It uses MemoryStore by default but users can configure it to use whatever they want.

That's probably the best if we pick a good generic default. I've done some work with Rails.cache with memcached and MemoryStore and saw some performance improvements.

#5 - 2010-02-06 11:33 - Jean-Philippe Lang

- Status changed from New to Closed

- Target version set to 1.0.0 (RC)

- Resolution set to Fixed

Feature added in r3372, see commit log for details.
As far as I can tell, it offers a great boost for large wiki pages.