Redmine - Defect #5642

Very slow data fetching on user activity request

2010-06-04 07:55 - Paolo Freuli

Status:	Closed	**Start date:**	2010-06-04
Priority:	Normal	**Due date:**	2013-01-22
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Resolution:	No feedback	**Affected version:**	0.9.4

Description

When clicking on a user link redmine takes a lot of time to return a result. When more than one user is working, redmine stops working and a sequence of mongrel time out exceptions arise.

As far as I may understand, the problem seems to be related to the
Redmine::Action fetcher trying to retrieve and order information.

Can something be done to resolve this?

Thanks

History

#1 - 2010-06-04 08:10 - Felix Schäfer

Wee need either an error trace with sufficient information or (even better) a way to reproduce your error. See SubmittingBugs for more info.

#2 - 2010-06-04 12:38 - Paolo Freuli

- File mongrel.log added

Felix Schäfer wrote:

> Wee need either an error trace with sufficient information or (even better) a way to reproduce your error. See SubmittingBugs for more info.

I hope to be more helpful:

When clicking on user link

```
<%=h role %>: <%= @users_by_role[role].sort.collect{|u| link_to_user u}.join(", ") %><br />
```

in ../projects/show.rhtml

Redmine hangs.
I am in tail on the production.log but I can't see any explicit new request.
The same behaviour clicking on a User link in the Issue Description.
Redmine is pointing to a large database (loaded with many changesests and wiki edits).
Maybe I am wrong, but the problem might be that the Redmine:Action fetcher is trying to retrieve and ordering anything.

Some details about the environment:

1. redmine 0.9.4;
2. bitnami stack (not using mysql, but postgresql);
3. same behaviour running with or without apache;
4. same behaviour in production or developmente mode;
5. note: I have tried to switch to mysql too => same behaviour
6. running on linux

At the end I made the following change in *user_controller* show method to allow people to work without having redmine freezed:

```
#events = Redmine::Activity::Fetcher.new(User.current, :author => @user).events(nil, nil, :limit => 10)
@events_by_day = []#events.group_by(&:event_date)
```

As far as I could understand (in the code), the

```
Redmine::Activity::Fetcher.new(User.current, :author => @user).events(nil, nil, :limit => 10)
```

call may be very expensive, and even reordering seems to be done not efficiently.

PS:
I love redmine (all the same).
Thank you for your great job.

#3 - 2010-06-21 14:03 - Felix Schäfer

- Category deleted (SCM)

Could you please tell us more about your environment? See SubmittingBugs.

#4 - 2010-06-21 14:04 - Felix Schäfer

Oh, sorry, you know the page already. Could you please tell us the versions of rails/ruby/rack at least, as well as what platform you are on?

#5 - 2010-06-21 16:20 - Paolo Freuli

Felix Schäfer wrote:

> Oh, sorry, you know the page already. Could you please tell us the versions of rails/ruby/rack at least, as well as what platform you are on?

1. Redmine 0.9.4-0 on 2010-05-07 (bitnami stack)
2. running on Ubuntu 6.06
3. with postgresql 8.1
4. subversion 1.3.2-3ubuntu2~dapper1

Let me know if you need more information.
Thank you.

#6 - 2010-06-21 19:37 - Felix Schäfer

Yes, the ruby and rails versions :-)

I had a look at the code and couldn't find anything strange. I'd say your db has problems coping with the size of the db (you said you have lots of stuff in the redmine db), any chance the memory/caches are low, or some migrations from redmine failed/aren't applied and you lack some indexes?

#7 - 2010-06-21 19:59 - Holger Just

Paolo, you are right, the Activity Fetcher is very inefficient. But it alone should not force a system to halt.

Could you please tell us a bit about the hardware environment you are running at. Specifically, CPU and memory.

Normally, I would propose the following checks and changes:

- Each rails process can only answer exactly one request at a time. Run more then one mongrel process using mongrel-cluster or switch to passenger. That way you can at least provide some concurrency so that a singlke long-running request does not kill the entire application.
  - On a standard install I would run about 3 mongrels, if you have sufficient memory. But you have to check your queue length. Maybe, two are sufficient, maybe you need more
- Make sure to tune your Postgres to the available memory. Most standard installs are tuned for a minimum of shared memory, so queries on large tables which do not fit into the configured shared memory have to be cached on disk which kills performance.
- Check your IO meters and CPU usage. Do you run into swap? Do you see unusual IO spikes? Is the disk array hard at work regarding IOPS and throughput? A standard hard disk can perform about 80-150 IOPS.
- Probably use a application query analyzer / monitor to find out where the time is spent. Newrelic RPM has been successfully used for that. They also have a free version.

#8 - 2010-06-22 14:34 - Paolo Freuli

Felix Schäfer wrote:

> Yes, the ruby and rails versions :-)
>
> I had a look at the code and couldn't find anything strange. I'd say your db has problems coping with the size of the db (you said you have lots of stuff in the redmine db), any chance the memory/caches are low, or some migrations from redmine failed/aren't applied and you lack some indexes?

ruby 1.8.7 (2010-01-10 patchlevel 249) [i686-linux]
rails (2.3.5)

Note (replying to indexes question):
I have the same problem both running with postgresql and with mysql (loaded with the same data) as reported in the issue description.

#9 - 2010-06-22 14:38 - Paolo Freuli

Holger Just wrote:

> Paolo, you are right, the Activity Fetcher is very inefficient. But it alone should not force a system to halt.
>
> Could you please tell us a bit about the hardware environment you are running at. Specifically, CPU and memory.
>
> Normally, I would propose the following checks and changes:
>
> - Each rails process can only answer exactly one request at a time. Run more then one mongrel process using mongrel-cluster or switch to passenger. That way you can at least provide some concurrency so that a singlke long-running request does not kill the entire application.
>   - On a standard install I would run about 3 mongrels, if you have sufficient memory. But you have to check your queue length. Maybe, two are sufficient, maybe you need more
> - Make sure to tune your Postgres to the available memory. Most standard installs are tuned for a minimum of shared memory, so queries on large tables which do not fit into the configured shared memory have to be cached on disk which kills performance.
> - Check your IO meters and CPU usage. Do you run into swap? Do you see unusual IO spikes? Is the disk array hard at work regarding IOPS and throughput? A standard hard disk can perform about 80-150 IOPS.
> - Probably use a application query analyzer / monitor to find out where the time is spent. Newrelic RPM has been successfully used for that. They also have a free version.

Hello Holger thank you for your reply!
I will check asap what you suggested.

FYI:

1. I am running a bitnami stack (mongrel_cluster -> two mongrels running)
2. hdd =5Gb
3. RAM =512Mb
4. behaviour is quite deterministic. The first (wrong) click causes the first mongrel process to halt, the second (wrong) click causes the second mongrel process to halt.

#10 - 2010-08-26 00:39 - Eric Davis

- Priority changed from Urgent to Normal

#11 - 2013-01-15 11:32 - Daniel Felix

- Due date set to 2013-01-22

- Status changed from New to Needs feedback

The affected version is very old and Redmine encountered many improvements on the way how database sets are retrieved.

Please give some feedback until next week, if this is still reproduceable or if this bug is already fixed during the past 2 years.

#12 - 2013-02-04 16:06 - Daniel Felix

- Status changed from Needs feedback to Closed

- Resolution set to No feedback

Closing this, as there is no feedback on this issue and the affected version is really old.

Files

mongrel.log	36.9 KB	2010-06-04	Paolo Freuli