Redmine - Defect #2824

MySQL Deadlock Error when showing a big repository

2009-02-24 14:27 - Minjie Zhu

Status:	New	**Start date:**	2009-02-24
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:	SCM	**Estimated time:**	0.00 hour
Target version:			
Resolution:		**Affected version:**	

Description

First thanks for your great jobs for this application.

I have some problem when I try to connect SVN with Redmine.

Here is the infomation of my system:

CentOS	5
MySQL	5.0.45
ruby	1.8.5
Rails	2.2.2
Redmine	r2521 (I also checked it with 0.8.0)
SVN	1.4.2

I've added a big SVN repository with 8k of revision and when I go to the repository tab, it show **internal error** after a long loading time.

I checked the log and it said

```
ActiveRecord::StatementInvalid (Mysql::Error: Deadlock found when trying to get lock; try restarti
ng transaction: INSERT INTO `changesets`...
```

The details of the trace is in the attachment.

Then I googled this problem, and it said this problem happen because of the MySQL InnoDB Deadlock detection,

Here is the details
http://dev.mysql.com/doc/refman/5.0/en/innodb-deadlock-detection.html

And here are some solutions
http://dev.mysql.com/doc/refman/5.0/en/innodb-deadlocks.html

I think the InnoDB has a time limit and if a transaction exceed it, the transaction will be aborted and rollbacked.

They said the application should retry the transaction or devide it to some short transactions.

There is a switch to on/off the function, and both the switch and the limit time are defined as system variable of MySQL which can be confirmed like this:

```
mysql> show variables like 'inno%';
```

You can find **innodb_lock_wait_timeout** which is the time limit with a default value 50 (seconds), and **innodb_table_locks** is the switch which the default value is ON.

I changed the value above by set them in /etc/my.cnf.
First I set the **innodb_lock_wait_timeout** to a very large number like '99999', but it didn't work. Then I set the **innodb_table_locks** to 'OFF', and it did work! I can view the repository tab though it still took a long time to load initially.

I don't know if there are some problems when I OFF the table lock function because it seem to be important for the DB and Redmine, Could you give me some solutions?

Sorry for my poor English and thanks for your help.

Related issues:

Related to Redmine - Defect #13086: "Mysql::Error: Duplicate entry" in fetchi...	**Needs feedback**	
Related to Redmine - Defect #13592: ActiveRecord::RecordNotUnique for git rep...	**New**	
Related to Redmine - Defect #13744: Redmine hangs up with large repo	**Closed**	
Related to Redmine - Defect #1931: Timeout risks: need asynchronous operations	**New**	**2008-09-22**

History

#1 - 2009-02-24 21:20 - Isaac Cruz Ballesteros

A deadlock has nothing to with lock wait timeout, a deadlock occurs when transaction A blocks transaction B, and B blocks A

Enter a 'show innodb status', it should describe the cause of the last deadlock

#2 - 2009-02-25 01:55 - Minjie Zhu

- File innodb_status_full.txt added

Isaac Cruz Ballesteros wrote:

> A deadlock has nothing to with lock wait timeout, a deadlock occurs when transaction A blocks transaction B, and B blocks A

Yes, I changed the lock wait time and it didn't work.

> Enter a 'show innodb status', it should describe the cause of the last deadlock

Here is the output of the command (a full output in attachment)

```
=====================================
090225  9:30:05 INNODB MONITOR OUTPUT
=====================================
Per second averages calculated from the last 11 seconds
----------
SEMAPHORES
----------
OS WAIT ARRAY INFO: reservation count 64, signal count 64
Mutex spin waits 0, rounds 870, OS waits 1
RW-shared spins 171, OS waits 13; RW-excl spins 66, OS waits 50
------------------------
LATEST DETECTED DEADLOCK
------------------------
090224 20:36:24
*** (1) TRANSACTION:
TRANSACTION 0 10412, ACTIVE 0 sec, process no 19408, OS thread id 3032349584 inserting
mysql tables in use 1, locked 1
LOCK WAIT 3 lock struct(s), heap size 320, undo log entries 1
MySQL thread id 5, query id 20932 localhost root update
INSERT INTO `changesets` (`commit_date`, `committed_on`, `comments`, `revision`, `scmid`, `user_id`, `reposito
ry_id`, `committer`) VALUES('2007-10-26 20:29:48', '2007-10-26 20:29:48', 'test.log', '601', NULL, NULL, 1, 't
anaka')
*** (1) WAITING FOR THIS LOCK TO BE GRANTED:
RECORD LOCKS space id 0 page no 281 n bits 944 index `changesets_repos_rev` of table `redmine/changesets` trx
id 0 10412 lock mode S waiting
Record lock, heap no 278 PHYSICAL RECORD: n_fields 3; compact format; info bits 0
 0: len 4; hex 80000001; asc     ;; 1: len 3; hex 363031; asc 601;; 2: len 4; hex 80000259; asc    Y;;

*** (2) TRANSACTION:
TRANSACTION 0 10411, ACTIVE 29 sec, process no 19408, OS thread id 3032550288 setting auto-inc lock
mysql tables in use 1, locked 1
4 lock struct(s), heap size 1024, undo log entries 2130
MySQL thread id 4, query id 20970 localhost root update
INSERT INTO `changesets` (`commit_date`, `committed_on`, `comments`, `revision`, `scmid`, `user_id`, `reposito
ry_id`, `committer`) VALUES('2007-10-30 13:50:20', '2007-10-30 13:50:20', 'add', '674', NULL, NULL, 1, 'suzuki
')
*** (2) HOLDS THE LOCK(S):
RECORD LOCKS space id 0 page no 281 n bits 944 index `changesets_repos_rev` of table `redmine/changesets` trx
```

```
id 0 10411 lock_mode X locks rec but not gap
Record lock, heap no 278 PHYSICAL RECORD: n_fields 3; compact format; info bits 0
 0: len 4; hex 80000001; asc     ;; 1: len 3; hex 363031; asc 601;; 2: len 4; hex 80000259; asc    Y;;

*** (2) WAITING FOR THIS LOCK TO BE GRANTED:
TABLE LOCK table `redmine/changesets` trx id 0 10411 lock mode AUTO-INC waiting
*** WE ROLL BACK TRANSACTION (1)
------------
TRANSACTIONS
------------
Trx id counter 0 10455
Purge done for trx's n:o < 0 10400 undo n:o < 0 0
History list length 2
Total number of lock structs in row lock hash table 0
LIST OF TRANSACTIONS FOR EACH SESSION:
---TRANSACTION 0 0, not started, process no 19408, OS thread id 3032750992
MySQL thread id 6, query id 166588 localhost root
show innodb status
```

#3 - 2009-02-25 10:42 - Jean-Philippe Lang

See the doc about repositories:

> **Important**: When you first browse the repository, Redmine retrieves the description of all of the existing commits and stores them in the database.
> This is done only once per repository but can very long (or even time out) if your repository has hundreds of commits.
>
> To avoid this, you can do it offline.
> After having your repository declared in Redmine, run the following command:
>
> ruby script/runner "Repository.fetch_changesets" -e production
>
> All commits will be retrieved in the Redmine database.

#4 - 2009-02-26 10:40 - Minjie Zhu

I know this way and it worked well, but in many case users cannot know how long the process will take until they try it, because it depends on the size of repository and the machine power. So I think users always try it by browser at first, but sometimes it will return an internal error which users don't know what caused it. That is why I think this is a defect of Redmine.

If, for example, the first fetch process ends with a timeout error, Redmine will detect it and tell user to do it offline instead. I think this will be a more friendly interface for user. Thank you.

#5 - 2012-10-31 22:05 - Daniel Felix

Well, this could be closed too, doesn't it?

It seems to be some system specific error. This seems to be caused by to much data to transfer and a to short time to do this. ;-)

#6 - 2012-11-01 01:16 - Toshi MARUYAMA

- Category changed from Database to SCM

- Priority changed from High to Low

#7 - 2012-11-01 01:32 - Toshi MARUYAMA

I think we cannot close this issue.
Redmine does not use lock in fetching revisions from SCMs.
To prevent long time fetching, in Git and Mercurial, you can **push** divided revisions to repository.

For example.

```
$ hg push -r 0.6.0
$ hg push -r 0.7.0
$ hg push -r 0.8.0
.
.
.
```

#8 - 2012-11-01 07:28 - Abdul Halim Mat Ali

Toshi MARUYAMA, this should be closed as it is not a bug.

I have a lot of repository, each with many revisions and over 200MB of data.

Hence, running

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

is critical after setting up the repository.

Once the script is run, it can take almost 1 hour, the next time you click the repository this issue will not happen.

#9 - 2015-05-21 19:37 - @ go2null

Related to #1931 *Timeout risks: need asynchronous operations*

Should be closed same as #7002 *mysql timeout when first browsing a large git repository*?

#10 - 2015-06-20 12:26 - Toshi MARUYAMA

- Related to Defect #1931: Timeout risks: need asynchronous operations added

Files

trace.log	10.7 KB	2009-02-24	Minjie Zhu
innodb_status_full.txt	19.1 KB	2009-02-25	Minjie Zhu