

projekt kivitendo - Fehler #180

Hänger / Verklemmung bei Benutzung von Rose und standard_dbh

22.06.2016 08:20 - Martin Helmling

Status:	Erledigt	Beginn:	04.06.2016
Priorität:	Hoch	Abgabedatum:	
Zugewiesen an:	Martin Helmling	% erledigt:	100%
Kategorie:		Geschätzter Aufwand:	30.00 Stunden
Zielversion:		Aufgewendete Zeit:	0.00 Stunde
Beschreibung			
Eigentlich gibt es das alte Ticket http://trac.kivitendo.de/ticket/2349 , das ein ähnliches Verhalten zeigt.			
Ich hab hier z.B. einen reproduzierbaren Fall bei dem zwei Testprozess gleichzeitig arbeiten. Der erste macht nur ein Update eines Auftrags Der zweite "Speichert als Neu" einen Artikel. Im log ist Log4perl eingeschaltet, in io.pl habe ich um make_record ein enter/leave debug .			
Erkennbar wird, dass der Updateprozess per standard_dbh noch parts offen hat, dann kommt der lock Request des TransNumber für den neuen Artikel, anschließend der Rose-Request auf parts im make_record, der dann blockiert wird.			
Anbei 4 Files: 2 perlscrippts zum Auslösen der Requests, 1 shellscrippt doit, dass die beiden Scripts in einer entsprechenden Zeitverzögerung laufen lasst (ist sicher Rechnerabhängig) 1 logfile mit der Verklemmung			
22111 pts/0 S 0:01 _ /usr/bin/perl scripts/testcmd1.pl			
22113 pts/0 S 0:01 _ /usr/bin/perl scripts/testcmd2.pl			

Historie

#1 - 22.06.2016 08:44 - Martin Helmling

- Datei log2 wurde hinzugefügt

Durch Schließen des standard_dbh vor Aufruf von Rose konnte dieser Fall entflochten werden (siehe log2):

```
sub _make_record {
+ Form::disconnect_standard_dbh;
  my $class = {
    sales_order => 'Order',
```

Doch sicher gibt es weitere solcher Fälle.

Wie kriegen wir das in den Griff?

#2 - 22.06.2016 08:55 - Martin Helmling

Mit Sven hatte ich diskutiert:

Warum geht es eigentlich nicht pauschal im Dispatcher bereits die standard-dbh auf die Rose-Verbindung zu setzen?
Damit würden Verklemmungen innerhalb eines Request ausgeschaltet sein,
also

```
$::form->set_standard_dbh(SL::DB::Object->new->db->dbh);
```

Weil die standard_dbh ohne autocommit ist, und die Roseverbindung mit autocommit ist. Du müsstest also alles umbauen, sodass die Transaktionen manuell >gestartet werden. Und ich vermute, dass werden wir auch irgendwann tun müssen.

Das ist aus meiner Sicht umgekehrt, da die Roseverbindung ein Autocommit hat ist kein commit mehr nötig, vielmehr müsste der rollback explizit an den Stellen gemacht werden wo er nötig ist

#3 - 23.06.2016 11:41 - Sven Schöling

Das ist aus meiner Sicht umgekehrt, da die Roseverbindung ein Autocommit hat ist kein commit mehr nötig, vielmehr müsste der rollback explizit an den Stellen gemacht werden wo er nötig ist.

Neee. Wir wollen ja, dass Sachen in Transaktionen ablaufen. Ansonsten kriegst Du so wunderschöne Effekte, dass Ein Auftrag gespeichert wird, und dann ein SQL Fehler irgendwo bei einer der Positionen passiert, und dann hast du einen halben Zombieauftrag in der Datenbank.

Ohne eine Transaktion ist ein rollback auch nur ein noop.

#4 - 24.06.2016 11:23 - Sven Schöling

Ich hab rausgefunden was da passiert Martin. Ich poste das auf die Mailingliste damit andere das auch lesen, und pack dann ne Kopie davon hier in den Bug.

#5 - 02.08.2016 10:14 - Martin Helmling

So nun habe ich auch sowas beim CSV-Importieren von vier Zeilen Artikeln, der Taskmanger hängt und wartet auf postgres

```
9408 ? Ss 0:00 \_ postgres: postgres kivitendo_auth_rnr ::1(54323) idle
9410 ? Ss 0:00 \_ postgres: postgres erp2014d04-rnr ::1(54325) LOCK TABLE waiting
9411 ? Ss 0:00 \_ postgres: postgres kivitendo_auth_rnr ::1(54326) idle
9412 ? Ss 0:00 \_ postgres: postgres erp2014d04-rnr ::1(54327) idle in transaction
```

relname	locktype	page	virtualtransaction	pid	mode	granted
bin	relation	11/15596		9410	AccessShareLock	t
custom_variable_configs	relation	11/15596		9410	RowShareLock	t
custom_variable_configs	relation	11/15596		9410	AccessShareLock	t
custom_variables	relation	11/15596		9410	AccessShareLock	t
custom_variables	relation	11/15596		9410	RowExclusiveLock	t
employee	relation	11/15596		9410	RowShareLock	t
history_erp	relation	11/15596		9410	RowExclusiveLock	t
parts	relation	11/15596		9410	RowExclusiveLock	t
parts	relation	11/15596		9410	AccessExclusiveLock	f
parts	relation	12/11557		9412	AccessShareLock	t
partstypes	relation	11/15596		9410	RowShareLock	t

#6 - 02.08.2016 15:27 - Martin Helmling

CSV-Import gefixed in 22cc0ebc5016c:

```
my $sth = prepare_execute_query($::form, $::form->get_standard_dbh, 'SELECT partnumber FROM parts');
my $sth = prepare_execute_query($::form, SL::DB::Object->new->db->dbh, 'SELECT partnumber FROM parts');
```

#7 - 12.01.2017 15:13 - Martin Helmling

- Status wurde von Neu zu Erledigt geändert

- % erledigt wurde von 60 zu 100 geändert

Nachdem die Single-dbh Änderungen beim Kunden laufen, wurden keine Verklemmungen mehr gemeldet

Dateien

Datei	Größe	Datum	Autor
log1	207 KB	22.06.2016	Martin Helmling
testcmd1.pl	7,03 KB	22.06.2016	Martin Helmling
testcmd2.pl	4,75 KB	22.06.2016	Martin Helmling
doit.sh	166 Bytes	22.06.2016	Martin Helmling
log2	364 KB	22.06.2016	Martin Helmling