Arvados - Bug #14966

[API] Fix hanging test - suspect permission changes

03/13/2019 06:07 PM - Tom Clegg

**Status:** Resolved  
**Priority:** Normal  
**Assigned To:** Lucas Di Pentima  
**Category:**  
**Target version:** 2019-03-27 Sprint

**Description**


...  
12:55:16 Arvados::V1::KeepDisksControllerTest#test_refuse_to_add_keep_disk_without_admin_token = 0
0.1 s = .  
12:55:16 Arvados::V1::KeepDisksControllerTest#test_search_keep_services_with_'any'_operator = 0.03 
[hangs here]

ps

```
postgres 15393  0.4  1.1  329040  91640 ?  Ss   16:49   0:18 postgres: 9.6/main: ci_test_user
arvados_test_api ::1(40670) REFRESH MATERIALIZED VIEW waiting  
postgres 15519  0.2  1.0  318256  82320 ?  Ss   16:50   0:08 postgres: 9.6/main: ci_test_user
arvados_test_api ::1(40706) idle in transaction  
postgres 15521  0.2  1.2  336216  92800 ?  Ss   16:50   0:08 postgres: 9.6/main: ci_test_user
arvados_test_api ::1(40708) PARSE waiting
```

Suspicious of a race condition in the async-permission-update changes recently merged in #13593.

**Subtasks:**

| Task # 14970: Review 14966-api-test-hang-fix | Resolved |

**Related issues:**

| Related to Arvados - Bug #13593: [API] Sequence of "create group" requests runs slowly, and can crash API server | Resolved | 03/12/2019 |

**Associated revisions**

Revision dc71f08c - 03/14/2019 07:42 PM - Lucas Di Pentima

Merge branch '14966-api-test-hang-fix'

Closes #14966

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

**History**

1 - 03/13/2019 06:07 PM - Tom Clegg

- Related to Bug #13593: [API] Sequence of "create group" requests runs slowly, and can crash API server added

2 - 03/13/2019 06:15 PM - Lucas Di Pentima

- Assigned To set to Lucas Di Pentima
- Target version set to 2019-03-27 Sprint

3 - 03/13/2019 06:20 PM - Lucas Di Pentima

- Status changed from New to In Progress

4 - 03/13/2019 11:58 PM - Lucas Di Pentima

Updates at 0e9a62c7 - branch 14966-api-test-hang-fix


- Facts
  - Minitest has a feature that’s active by default called 'transactional fixtures' that encloses every test in a transaction so it’s rolled back before

05/07/2021
Nested transactions share the same database connection ([https://apidock.com/rails/ActiveRecord/ConnectionAdapters/DatabaseStatements/transaction](https://apidock.com/rails/ActiveRecord/ConnectionAdapters/DatabaseStatements/transaction)). The permissions graph update is run inside a transaction. 

**Hypothesis**

When a permission-changing operation is made within a test, we’re getting nested transactions that share the same database connection. This is usually fine because both run within the same thread, but for this special case that we expect the inner transaction to run concurrently in its own thread, it doesn’t happen because the connection is already occupied and sits waiting to be released before continuing. 

**Fix**

Although I already tried to manually ask for a new connection using ActiveRecord::Base.connection_pool.with_connection do ... end on refresh_permission_view(), it seems there’s no obvious way to ask for a new connection so instead the proposed fix would be to move this special test on its own test case class, disabling transactional fixtures. The “cons” that this has is that the changes made on this test would leak to other tests, so we have to make sure to clean up before finishing.

#5 - 03/14/2019 05:00 PM - Peter Amstutz

Lucas Di Pentima wrote:

Updates at 0e9aa62c7 - branch 14966-api-test-hang-fix

- **Facts**
  - Minitest has a feature that’s active by default called ‘transactional fixtures’ that encloses every test in a transaction so it’s rolled back before running the next.
  - Nested transactions share the same database connection ([https://apidock.com/rails/ActiveRecord/ConnectionAdapters/DatabaseStatements/transaction](https://apidock.com/rails/ActiveRecord/ConnectionAdapters/DatabaseStatements/transaction))
  - The permissions graph update is run inside a transaction

- **Hypothesis**
  - When a permission-changing operation is made within a test, we’re getting nested transactions that share the same database connection. This is usually fine because both run within the same thread, but for this special case that we expect the inner transaction to run concurrently in its own thread, it doesn’t happen because the connection is already occupied and sits waiting to be released before continuing.

- **Fix**
  - Although I already tried to manually ask for a new connection using ActiveRecord::Base.connection_pool.with_connection do ... end on refresh_permission_view(), it seems there’s no obvious way to ask for a new connection so instead the proposed fix would be to move this special test on its own test case class, disabling transactional fixtures.
  - The “cons” that this has is that the changes made on this test would leak to other tests, so we have to make sure to clean up before finishing.

I'm not sure I totally understand the explanation, but agree that transactions / connection pooling could be causing trouble. Since it is an integration test it seems reasonable to run the test without the an outer transaction. Couple notes:

The test should be wrapped in a begin ... ensure ... end.

To reset the database, suggest using the database reset endpoint:

```
post '/database/reset', {}, admin_auth
assert_response :success
```

typo "accesible" -> "accessible"

#6 - 03/14/2019 06:58 PM - Lucas Di Pentima

Updates at 0145b3654

Instead of resetting the database inside an ensure block, do it at test case’s teardown.

#7 - 03/14/2019 07:57 PM - Lucas Di Pentima

- Status changed from In Progress to Resolved
- % Done changed from 0 to 100

Applied in changeset arvadosidec71f08c4a79e653a4da6f8fd054a9e39600ad.

#8 - 05/21/2019 10:27 PM - Tom Morris

- Release set to 15