Pulp - Issue #7073

import test fails with psycopg2.extensions.TransactionRollbackError: deadlock detected

07/01/2020 08:09 PM - dkliban@redhat.com

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	ggainey	**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:	3.5.0		
Severity:	2. Medium	**Groomed:**	No
Version:		**Sprint Candidate:**	No
Platform Release:		**Tags:**	
OS:		**Sprint:**	Sprint 76
Triaged:	No	**Quarter:**	

Description

The test_import test is failing intermittently on Travis:

```
=================================== FAILURES ===================================
________________________ PulpImportTestCase.test_import ________________________
self = 
    def test_import(self):
        """Test an import."""
        importer = self._create_importer()
>       task_group = self._perform_import(importer)
pulpcore/tests/functional/api/using_plugin/test_pulpimport.py:194: 
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
pulpcore/tests/functional/api/using_plugin/test_pulpimport.py:167: in _perform_import
    task_group = monitor_task_group(task_group_href)
pulpcore/tests/functional/utils.py:54: in monitor_task_group
    exit()
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
self = Use exit() or Ctrl-D (i.e. EOF) to exit, code = None
    def __call__(self, code=None):
        # Shells like IDLE catch the SystemExit, but listen when their
        # stdin wrapper is closed.
        try:
            sys.stdin.close()
        except:
            pass
>       raise SystemExit(code)
E       SystemExit: None
../../../virtualenv/python3.7.6/lib/python3.7/site.py:403: SystemExit
----------------------------- Captured stdout call -----------------------------
The task group failed: {'all_tasks_dispatched': True,
 'canceled': 0,
 'completed': 1,
 'description': 'Import of '
                '/tmp/0cf747ab-d5b3-454a-88f0-7aedf5fad6f3/export-5a84c6c2-4905-4400-b2ea-f48e8c603e8b-20200701_1100.tar.gz',
 'failed': 1,
 'group_progress_reports': [],
 'pulp_href': '/pulp/api/v3/task-groups/a8fc525b-eb7d-442d-a732-694648485198/',
 'running': 0,
 'skipped': 0,
 'waiting': 0}
```

The following logs appear.

```
pulp: rq.worker:ERROR: django.db.utils.OperationalError: deadlock detected
DETAIL:  Process 4137 waits for ShareLock on transaction 6954; blocked by process 4128.
Process 4128 waits for ShareLock on transaction 6951; blocked by process 4137.
HINT:  See server log for query details.
CONTEXT:  while updating tuple (0,29) in relation "core_content"
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 84, in _execute
    return self.cursor.execute(sql, params)
psycopg2.extensions.TransactionRollbackError: deadlock detected
DETAIL:  Process 4137 waits for ShareLock on transaction 6954; blocked by process 4128.
Process 4128 waits for ShareLock on transaction 6951; blocked by process 4137.
HINT:  See server log for query details.
CONTEXT:  while updating tuple (0,29) in relation "core_content"
The above exception was the direct cause of the following exception:
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/rq/worker.py", line 883, in perform_job
    rv = job.perform()
  File "/usr/local/lib/python3.7/site-packages/rq/job.py", line 657, in perform
    self._result = self._execute()
  File "/usr/local/lib/python3.7/site-packages/rq/job.py", line 663, in _execute
    return self.func(*self.args, **self.kwargs)
  File "/usr/local/lib/python3.7/site-packages/pulpcore/app/tasks/importer.py", line 126, in import_repository_version
    c_result = _import_file(content_path, ContentResource)
  File "/usr/local/lib/python3.7/site-packages/pulpcore/app/tasks/importer.py", line 47, in _import_file
    return resource.import_data(data, raise_errors=True)
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 627, in import_data
    return self.import_data_inner(dataset, dry_run, raise_errors, using_transactions, collect_failed_rows, **kwargs)
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 673, in import_data_inner
    raise row_result.errors[-1].error
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 569, in import_row
    self.save_instance(instance, using_transactions, dry_run)
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 352, in save_instance
    instance.save()
  File "/usr/local/lib/python3.7/site-packages/pulpcore/app/models/base.py", line 110, in save
    return super().save(*args, **kwargs)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 741, in save
    force_update=force_update, update_fields=update_fields)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 779, in save_base
    force_update, using, update_fields,
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 851, in _save_table
    forced_update)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 900, in _do_update
    return filtered._update(values) > 0
  File "/usr/local/lib/python3.7/site-packages/django/db/models/query.py", line 760, in _update
    return query.get_compiler(self.db).execute_sql(CURSOR)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/sql/compiler.py", line 1469, in execute_sql
    cursor = super().execute_sql(result_type)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/sql/compiler.py", line 1140, in execute_sql
    cursor.execute(sql, params)
  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 67, in execute
    return self._execute_with_wrappers(sql, params, many=False, executor=self._execute)
  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 76, in _execute_with_wrappers
    return executor(sql, params, many, context)
  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 84, in _execute
    return self.cursor.execute(sql, params)
  File "/usr/local/lib/python3.7/site-packages/django/db/utils.py", line 89, in __exit__
    raise dj_exc_value.with_traceback(traceback) from exc_value
  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 84, in _execute
    return self.cursor.execute(sql, params)
django.db.utils.OperationalError: deadlock detected
DETAIL:  Process 4137 waits for ShareLock on transaction 6954; blocked by process 4128.
Process 4128 waits for ShareLock on transaction 6951; blocked by process 4137.
HINT:  See server log for query details.
CONTEXT:  while updating tuple (0,29) in relation "core_content"
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 84, in _execute
    return self.cursor.execute(sql, params)
psycopg2.extensions.TransactionRollbackError: deadlock detected
DETAIL:  Process 4137 waits for ShareLock on transaction 6954; blocked by process 4128.
Process 4128 waits for ShareLock on transaction 6951; blocked by process 4137.
HINT:  See server log for query details.
CONTEXT:  while updating tuple (0,29) in relation "core_content"
The above exception was the direct cause of the following exception:
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/site-packages/rq/worker.py", line 883, in perform_job
    rv = job.perform()
  File "/usr/local/lib/python3.7/site-packages/rq/job.py", line 657, in perform
```

```
    self._result = self._execute()
  File "/usr/local/lib/python3.7/site-packages/rq/job.py", line 663, in _execute
    return self.func(*self.args, **self.kwargs)
  File "/usr/local/lib/python3.7/site-packages/pulpcore/app/tasks/importer.py", line 126, in import_repository_version
    c_result = _import_file(content_path, ContentResource)
  File "/usr/local/lib/python3.7/site-packages/pulpcore/app/tasks/importer.py", line 47, in _import_file
    return resource.import_data(data, raise_errors=True)
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 627, in import_data
    return self.import_data_inner(dataset, dry_run, raise_errors, using_transactions, collect_failed_rows, **kwargs)
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 673, in import_data_inner
    raise row_result.errors[-1].error
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 569, in import_row
    self.save_instance(instance, using_transactions, dry_run)
  File "/usr/local/lib/python3.7/site-packages/import_export/resources.py", line 352, in save_instance
    instance.save()
  File "/usr/local/lib/python3.7/site-packages/pulpcore/app/models/base.py", line 110, in save
    return super().save(*args, **kwargs)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 741, in save
    force_update=force_update, update_fields=update_fields)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 779, in save_base
    force_update, using, update_fields,
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 851, in _save_table
    forced_update)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/base.py", line 900, in _do_update
    return filtered._update(values) > 0
  File "/usr/local/lib/python3.7/site-packages/django/db/models/query.py", line 760, in _update
    return query.get_compiler(self.db).execute_sql(CURSOR)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/sql/compiler.py", line 1469, in execute_sql
    cursor = super().execute_sql(result_type)
  File "/usr/local/lib/python3.7/site-packages/django/db/models/sql/compiler.py", line 1140, in execute_sql

    cursor.execute(sql, params)

  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 67, in execute

    return self._execute_with_wrappers(sql, params, many=False, executor=self._execute)

  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 76, in _execute_with_wrappers

    return executor(sql, params, many, context)

  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 84, in _execute

    return self.cursor.execute(sql, params)

  File "/usr/local/lib/python3.7/site-packages/django/db/utils.py", line 89, in __exit__

    raise dj_exc_value.with_traceback(traceback) from exc_value

  File "/usr/local/lib/python3.7/site-packages/django/db/backends/utils.py", line 84, in _execute

    return self.cursor.execute(sql, params)

django.db.utils.OperationalError: deadlock detected

DETAIL:  Process 4137 waits for ShareLock on transaction 6954; blocked by process 4128.

Process 4128 waits for ShareLock on transaction 6951; blocked by process 4137.

HINT:  See server log for query details.

CONTEXT:  while updating tuple (0,29) in relation "core_content"
```

There is a theory behind such a deadlock in this stack overflow post: https://stackoverflow.com/a/42731706

Related issues:

Related to RPM Support - Issue #7560: Import tests are failing with psycopg2.... **NEW**

Associated revisions

Revision 94cf0081 - 07/02/2020 09:04 PM - ggainey

Prevents deadlock by locking core_content across update.

THIS IS A TEMPORARY FIX to prevent test_pulpimport failures from blocking unrelated commits. See the related issue for other approaches.

Success can be tested in a dev-env with the following bash cmd:

for i in {1..10}; do (pytest -v -r sx --color=yes --pyargs
pulpcore.tests.functional.api.using_plugin.test_pulpimport > $i.out &) done

Without this fix, each run sees 1-4 tests fail due to deadlock. With the fix, repeated executions have shown no failures.

fixes #7073

History

#1 - 07/01/2020 08:09 PM - dkliban@redhat.com

- Description updated

#2 - 07/01/2020 08:26 PM - daviddavis

- Status changed from NEW to ASSIGNED

- Assignee set to daviddavis

#3 - 07/01/2020 08:26 PM - daviddavis

- Sprint set to Sprint 76

#4 - 07/01/2020 08:28 PM - pulpbot

- Status changed from ASSIGNED to POST

PR: https://github.com/pulp/pulpcore/pull/768

#5 - 07/01/2020 10:29 PM - ggainey

I forced a deadlock by running test_pulpimport in parallel :

```
for i in {1..10}; do (pytest -v -r sx --color=yes --pyargs pulpcore.tests.functional.api.using_plugin.test_pul
pimport > $i.out &); done
```

One test failed of the ten. If I can make it happen reliably, I'll be able to test daviddavis' proposed PR and see how it affects the problem.

#6 - 07/01/2020 11:02 PM - ggainey

Transaction-isolation had no impact, lots of deadlocks.

The code in https://github.com/pulp/pulpcore/pull/745 does NOT deadlock - most likely because that change no longer explicitly import/exports core_content, which is the problem here. Investigation continues.

#7 - 07/01/2020 11:04 PM - ggainey

- Status changed from POST to ASSIGNED

- Assignee changed from daviddavis to ggainey

Moved back to assigned since the linked PR didn't fix the prob

#8 - 07/02/2020 02:32 AM - ggainey

The following diff fixes the deadlock by locking the entire table before the offending _import_file - but it is definitely a (horribly-nonperformant) "big hammer" approach. This could be generalized and moved inside _import_file() - but locking every table we touch at import would lose us most of the bandwidth we're getting by parallelizing repo-import in the first place. This ijust a step in the right direction:

```
diff --git a/pulpcore/app/tasks/importer.py b/pulpcore/app/tasks/importer.py
index a8c822b05..aa711af53 100644
--- a/pulpcore/app/tasks/importer.py
```

```
+++ b/pulpcore/app/tasks/importer.py
@@ -123,7 +123,14 @@ def import_repository_version(destination_repo_pk, source_repo_pk, tar_path):

         # Untyped Content
         content_path = os.path.join(rv_path, CONTENT_FILE)
-        c_result = _import_file(content_path, ContentResource)
+        from django.db import (
+            connection,
+            transaction,
+        )
+        with transaction.atomic():
+            cursor = connection.cursor()
+            cursor.execute('LOCK TABLE %s IN %s MODE' % (Content._meta.db_table, "EXCLUSIVE"))
+            c_result = _import_file(content_path, ContentResource)
         content = Content.objects.filter(pk__in=[r.object_id for r in c_result.rows])

         # Content Artifacts
```

#9 - 07/02/2020 03:06 AM - ggainey

Another option is to insure ordering on Content at export-time, by guaranteeing order at the Model level:

```
(master) ~/github/Pulp3/pulpcore $ git diff
diff --git a/pulpcore/app/models/content.py b/pulpcore/app/models/content.py
index 3f8d59890..e7eecfe24 100644
--- a/pulpcore/app/models/content.py
+++ b/pulpcore/app/models/content.py
@@ -260,6 +260,7 @@ class Content(MasterModel, QueryMixin):
     class Meta:
         verbose_name_plural = "content"
         unique_together = ()
+        ordering = ('pulp_id',)

     @classmethod
     def natural_key_fields(cls):

diff --git a/pulpcore/plugin/repo_version_utils.py b/pulpcore/plugin/repo_version_utils.py
index 70d7eb39c..e9052eaaa 100644
--- a/pulpcore/plugin/repo_version_utils.py
+++ b/pulpcore/plugin/repo_version_utils.py
@@ -93,6 +93,7 @@ def validate_duplicate_content(version):
         ).count()
         unique_new_content_total = (
             type_obj.objects.filter(pk__in=version.content.filter(pulp_type=pulp_type))
+            .order_by(*repo_key_fields)
             .distinct(*repo_key_fields)
             .count()
         )
```

That works to prevent the deadlock as well.

(Note the change to repo_version_utils is required, because distinct(keys) must match the ordering on the field. Since we've added a default ordering, which we're overriding, we must explicitly state the order_by() to avoid the dreaded error

```
ProgrammingError: SELECT DISTINCT ON expressions must match initial ORDER BY expressions
```

So - we now have three options:

1. Explicitly lock the table before operating on it.
2. Have an explicit default-order, to avoid the arbitrary-order-on-select postgres behavior that is the root of this issue.
3. Stop working with that table

That last just prevents this specific deadlock. Of the three, having an explicit default-ordering **for every Model** is the safest thing to do for the health of Pulp3 over time. It comes, of course, with a performance price. The deadlocks it will prevent are "multiple threads accessing multiple intersecting Model-instances in the same Model-class, while attempting to update the Models". Without explicitly ordering such requests, this is just a deadly embrace waiting to happen. And the more multi-user, and busier, a Pulp instance is, the more likely we are to eventually start seeing this become a problem.

#10 - 07/02/2020 07:20 PM - pulpbot

- Status changed from ASSIGNED to POST

PR: https://github.com/pulp/pulpcore/pull/769

#11 - 07/02/2020 09:04 PM - ggainey

- Status changed from POST to MODIFIED

Applied in changeset pulpcore|94cf00817669a4e7861806e85d3a252df4c16964.

#12 - 07/08/2020 09:00 PM - fao89

- Sprint/Milestone set to 3.5.0

#13 - 07/09/2020 05:06 PM - pulpbot

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE

#14 - 09/23/2020 08:55 PM - ttereshc

- Related to Issue #7560: Import tests are failing with psycopg2.extensions.TransactionRollbackError: deadlock detected added