Hi,

We're currently running Foreman 1.11.2 and it's impossible for us to expire Puppet reports using the standard way to do it (via the Rake task `reports:expire`) as the SQL queries that it issues are slow and generate locks in the reports table. This situation slows down our whole Puppet infrastructure as the masters start to wait too much for Foreman to process reports that they don't have time to digest new requests. This increments the backlog, increasing the catalog compilation time and the latency of basically all the requests that the agents perform.

We have a query killer in place in the database (a procedure, basically) that kills queries longer than 5 minutes, therefore every night the rake task is killed and no reports are cleaned-up. We do this essentially to protect our Puppet infra.

The query itself that's killed is the following:

```sql
DELETE FROM `logs` WHERE `logs`.'id' IN (SELECT id FROM (SELECT `logs`.'id' FROM `logs` INNER JOIN `reports` ON `reports`.'id` = `logs`.'report_id' WHERE `logs`.'report_id' IN (SELECT `reports`.'id' FROM `reports` WHERE `reports`.'type' IN ('ConfigReport')) AND (reports.created_at < '2016-07-04 03:30:08')) ORDER BY logs.id) __active_record_temp
```

which I guess comes from:

```
Log.joins(:report).where(:report_id => where(cond)).delete_all (app/models/report.rb:L82)
```

Our current workaround is the following:

1) Disable the query killer
2) Initiate a transaction and use a simplified query:
   2.1) START TRANSACTION
   2.2) DELETE FROM `logs` WHERE `logs`.'report_id' IN (SELECT `reports`.'id' FROM `reports` WHERE `reports`.'type' IN ('ConfigReport') AND (reports.created_at < '2016-07-04 03:30:08'))
   2.3) COMMIT the transaction (this does not generate any lock until the transaction can be committed and the performance is not degraded during the 'preparing' stage)
3) Now that all the logs of expired exports are deleted, execute again `reports:expire` so it has less work to do. This normally succeeds and creates no impact.
4) Enable the query killer

With this strategy we can kind of work around the problem but, can you think of any way to make it more efficient (and less harmless) to the application to expire reports using the Rake task?

Another option that we have is to declare a downtime every time that we expire reports but we'd rather not to go for this :)

03/28/2022
The rake task is triggered by a nightly cron in our case:

```
30 5 * * * (/usr/sbin/foreman-rake reports:expire) >> /var/log/foreman/reports-expire.log 2>&1
```

We're adding ~10 million entries to 'logs' per day and 340k to 'reports'. We're currently unable to expire 24 hours worth of reports without putting the infrastructure at risk.

Thanks for your help and time!

### Associated revisions

**Revision 65f95cab - 07/21/2016 03:53 AM - Tomer Brisker**

Fixes #15675 - Improve report expiry performance

Slowness was being caused by an un-needed `.joins` method, which led rails to creating suboptimal queries. Also removed default ordering for a bit of extra performance.

I tested on a largeish DB with 3M reports and 6M logs, time was cut by a factor of 13. See the PR for query plan comparison.

(Cherry picked from commit 65f95cab23d2e92439c20cf984ed38fad5542eff)

**Revision 1d8ada82 - 07/22/2016 10:49 AM - Tomer Brisker**

Fixes #15675 - Improve report expiry performance

Slowness was being caused by an un-needed `.joins` method, which led rails to creating suboptimal queries. Also removed default ordering for a bit of extra performance.

I tested on a largeish DB with 3M reports and 6M logs, time was cut by a factor of 13. See the PR for query plan comparison.

**History**

**#1 - 07/13/2016 08:47 AM - Dominic Cleal**
- Category set to Performance

**#2 - 07/20/2016 04:32 AM - Tomer Brisker**
- Status changed from New to Assigned
- Assignee set to Tomer Brisker

**#3 - 07/21/2016 04:01 AM - Anonymous**
- Status changed from Assigned to Closed
- % Done changed from 0 to 100
#4 - 07/21/2016 04:14 AM - Dominic Cleal
- Legacy Backlogs Release (now unused) set to 161

#5 - 07/21/2016 10:40 AM - Daniel Lobato Garcia
- Pull request https://github.com/theforeman/foreman/pull/3661 added
No idea what happened but the PR was not linked to the issue this time

#6 - 07/29/2016 01:21 PM - Daniel Lobato Garcia
- Target version set to 1.7.1

#7 - 08/08/2016 10:37 AM - Daniel Lobato Garcia
- Target version deleted (1.7.1)

#8 - 08/08/2016 10:44 AM - Daniel Lobato Garcia
- Target version set to 1.7.1

#9 - 08/23/2016 10:00 AM - Daniel Lobato Garcia
- Target version changed from 1.7.1 to 1.6.2

#10 - 08/23/2016 11:05 AM - Daniel Lobato Garcia
- Target version changed from 1.6.2 to 1.7.1