

Foreman - Refactor #8798

Reports.expire uses an unnecessary join on logs

12/28/2014 09:35 AM - Ori Rabin

Status: New	
Priority: Normal	
Assignee:	
Category: Database	
Target version:	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
When deleting the logs the line <code>Log.joins(:report)</code> doesn't need the reports table: https://github.com/theforeman/foreman/blob/d50c79908831be65da0b0005e438c9e819db613a/app/models/report.rb#L122	
In mysql the join uses the index and removing it causes the query to take longer. This needs to be checked in other dbs and refactored to use the index without the join.	
Related issues:	
Related to Foreman - Bug #8565: report::expire is running very slowly	Closed 12/03/2014

History

#1 - 01/02/2015 05:04 AM - Dominic Cleal

- Category changed from Reporting to Database

#2 - 01/21/2015 08:07 AM - Martin Jackson

Our experience is that the reports:expire rake task can get to where it can't finish with large numbers of reports to expire (we have 34,000 nodes and a two-hour runinterval); possibly because of this query. We were running PG 8.4; we upgraded to 9.2 and while the database itself is performing much better, we had to kill the reports:expire task after 2.5 hours because the server started swapping.

#3 - 01/21/2015 08:11 AM - Dominic Cleal

- Related to Bug #8565: report::expire is running very slowly added

#4 - 01/21/2015 08:41 AM - Ohad Levy

is this on 1.7? if you remove the join does it perform better? e.g.

```
--- a/app/models/report.rb
+++ b/app/models/report.rb
@@ -119,7 +119,7 @@ class Report < ActiveRecord::Base
   cond = "reports.created_at < \#{(Time.now.utc - timerange).to_formatted_s(:db)}\"
   cond += " and reports.status = #{status}" unless status.nil?

-   Log.joins(:report).where(:report_id => Report.where(cond)).delete_all
+   Log.where(:report_id => Report.where(cond)).delete_all
   Message.where("id not IN (#{Log.select(:message_id).to_sql})").delete_all
   Source.where("id not IN (#{Log.select(:source_id).to_sql})").delete_all
   count = Report.where(cond).delete_all
```