Foreman - Bug #24712

Foreman (1.18?) slow API call against fact_values endpoint

08/24/2018 01:32 PM - Luke Alexander

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Tomer Brisker</td>
</tr>
<tr>
<td>Category:</td>
<td>API</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Difficulty:</td>
<td></td>
</tr>
<tr>
<td>Triaged:</td>
<td>No</td>
</tr>
<tr>
<td>Bugzilla link:</td>
<td>1622343</td>
</tr>
<tr>
<td>Fixed in Releases:</td>
<td>1.18.2, 1.19.0</td>
</tr>
<tr>
<td>Found in Releases:</td>
<td>1.18.0, 1.18.1, 1.19.0-RC1, 1.19.0-RC2, 1.19.0-RC3</td>
</tr>
</tbody>
</table>

Description

Problem:
Foreman API call after upgrade now takes over 8 seconds which previously was under 1 second, I'm not sure if it's isolated to the fact_values endpoint, yet.

Tomer Brisker asked me to link a ref to another ticket which he thinks is the cause of this issue: https://projects.theforeman.org/issues/21871

Expected outcome:
Previous releases (1.16.*) returned the results in under 1 second

Foreman and Proxy versions:

Debian Stretch with:
ii foreman 1.18.1-1 amd64 Systems management web interface
ii foreman-cli 1.18.1-1 all metapackage providing hammer CLI for Foreman
ii foreman-debug 1.18.1-1 all provides support utility foreman-debug.
ii foreman-libvirt 1.18.1-1 all metapackage providing libvirt dependencies for Foreman
ii foreman-postgresql 1.18.1-1 all metapackage providing PostgreSQL dependencies for Foreman
ii foreman-proxy 1.18.1-1 all RESTful proxies for DNS, DHCP, TFTP, and Puppet
ii ruby-foreman-deface 1.2.0-1 all Foreman Deface Plugin Dependency
ii ruby-foreman-dhcp-browser 0.0.8-1 all Foreman DHCP browser Plugin
ii ruby-foreman-discovery 12.0.2-1 all Foreman Discovery Plugin
ii ruby-foreman-hooks 0.3.14-1 all Foreman Hooks Plugin
ii ruby-foreman-templates 6.0.3-1 all Foreman Templates Plugin
ii ruby-hammer-cli-foreman 0.13.0-1 all Foreman commands for Hammer

Other relevant data:

Example curl command:

time curl -k -u admin:admin -H "Accept: version=2, application/json" -H "Content-Type: application/json" -X GET "https://theforeman.example.net/api/fact_values?per_page=5&search=name=ipmi_macaddress" { 
  "total": 432656,
  "subtotal": 5,
  "page": 1,
  "per_page": 5,
  "search": "name=ipmi_macaddress",
  "sort": {
    "by": null,
    "order": null
  },

03/28/2022 1/4
real 0m0.244s
user 0m0.028s
sys 0m0.000s

I had some issues getting SQL debug logging working - but when I did, I got this in the logs (I snipped the list of IDs in the last query as it was many, many lines!)

```ruby
2018-08-22T10:02:45 [l@app][ddca7] Processing by Api::V2::FactValuesController#index as JSON
2018-08-22T10:02:45 [l@app][ddca7] Parameters: {
  "per_page"=>5,
  "search"=>"name=ipmi_macaddress",
  "apiv"=>"v2",
  "fact_value"=>[]
}
2018-08-22T10:02:45 [D][sql][ddca7] Setting Load (0.6ms) SELECT "settings".* FROM "settings" WHERE "settings".name = $1 ORDER BY "settings".name ASC LIMIT $2 [["name", "authenticate_login_delegation_api"], ["LIMIT", 1]]
2018-08-22T10:02:45 [D][sql][ddca7] User Load (0.3ms) SELECT "users".* FROM "users" WHERE "users".lower_login = $1 LIMIT $2 [["lower_login", "admin"], ["LIMIT", 1]]
2018-08-22T10:02:45 [D][sql][ddca7] PersonalAccessToken Load (0.5ms) SELECT "personal_access_tokens".* FROM "personal_access_tokens" WHERE "personal_access_tokens".revoked = $1 AND (expires_at = $2 OR expires_at IS NULL) AND "personal_access_tokens".user_id = 3 AND "personal_access_tokens".token = $3 LIMIT $4 [["revoked", "true"], ["token", "7ea0bb43659d7fe52511a090115b304b0fa14fe"], ["LIMIT", 1]]
2018-08-22T10:02:45 [D][sql][ddca7] AuthSource Load (0.5ms) SELECT "auth_sources".* FROM "auth_sources" WHERE "auth_sources".id = $1 LIMIT $2 [["id", 1], ["LIMIT", 1]]
2018-08-22T10:02:45 [D][sql][ddca7] User Load (0.3ms) SELECT "users".* FROM "users" WHERE "users".lower_login = $1 LIMIT $2 [["lower_login", "admin"], ["LIMIT", 1]]
2018-08-22T10:02:45 [D][sql][ddca7] Authenticated user admin against INTERNAL authentication source
2018-08-22T10:02:45 [D][sql][ddca7] User Load (0.3ms) SELECT "users".* FROM "users" WHERE "users".lower_login = $1 LIMIT $2 [["lower_login", "foreman_admin"], ["LIMIT", 1]]
2018-08-22T10:02:45 [l@app][ddca7] Current user: foreman_admin (administrator)
2018-08-22T10:02:45 [D][sql][ddca7] (0.2ms) SELECT auth_sources.id FROM "auth_sources" WHERE auth_sources.type IN ('AuthSourceHidden')
2018-08-22T10:02:45 [D][sql][ddca7] User Load (0.3ms) SELECT "users".* FROM "users" WHERE ('users'.auth_source_id NOT IN (2)) AND "users".lower_login = $1 LIMIT $2 [["lower_login", "admin"], ["LIMIT", 1]]
2018-08-22T10:02:45 [l@app][ddca7] Authorized user admin(Admin User)
2018-08-22T10:02:45 [l@app][ddca7] Current user: admin (administrator)
2018-08-22T10:02:45 [D][sql][ddca7] (0.1ms) BEGIN
2018-08-22T10:02:45 [D][sql][ddca7] SQL (0.2ms) DELETE FROM "sessions" WHERE "sessions".id = $1 ["id", 21634422]
2018-08-22T10:02:45 [D][sql][ddca7] (0.3ms) COMMIT
2018-08-22T10:02:45 [D][sql][ddca7] Setting Load (0.5ms) SELECT "settings".* FROM "settings" WHERE "settings".name = $1 ORDER BY "settings".name ASC LIMIT $2 [["name", "idle_timeout"], ["LIMIT", 1]]
2018-08-22T10:02:45 [D][sql][ddca7] SQL (3.6ms) SELECT "fact_values".* AS t0_r0, "fact_values".value AS t0_r1, "fact_value".fact_name_id AS t0_r2, "fact_values".host_id AS t0_r3, "fact_values".value AS t0_r4, "fact_values".created_at AS t0_r5, "fact_values".id AS t1_r0, "fact_names".name AS t1_r1, "fact_names".short_name AS t1_r2, "fact_names".type AS t1_r3, "fact_names".compose AS t1_r4, "fact_names".ancestry AS t1_r5, "fact_names".name AS t1_r6 FROM "fact_names" INNER JOIN "fact_values" ON "fact_names".id = "fact_values".fact_name_id WHERE (fact_names.name = '_timestamp') AND ("fact_names".name = 'ipmi_macaddress') ORDER BY "fact_values".value ASC NULLS FIRST $1 OFFSET $2 [["LIMIT", 5], ["OFFSET", 0]]
2018-08-22T10:02:46 [D][sql][ddca7] Host::Base Load (0.5ms) SELECT "hosts".* FROM "hosts" WHERE "hosts".id IN (1218, 517, 612, 687, 566)
2018-08-22T10:02:46 [D][sql][ddca7] Host::Managed Load (0.4ms) SELECT "hosts".* FROM "hosts" WHERE "hosts".type IN ('Host::Managed') AND "hosts".id IN (1218, 612, 687, 566)
2018-08-22T10:02:46 [l@app][ddca7] Rendering api/v2/fact_values/index.json.rabl within api/v2/layouts/index_layout
2018-08-22T10:02:46 [l@app][ddca7] Rendering api/v2/fact_values/index.json.rabl within api/v2/layouts/index_layout (0.8ms)
2018-08-22T10:02:46 [D][sql][ddca7] (135.5ms) SELECT "fact_values".* FROM "fact_values"
2018-08-22T10:02:46 [D][sql][ddca7] CACHE (0.0ms) SELECT "fact_values".* FROM "fact_values"
2018-08-22T10:02:54 [D][sql][ddca7] (2122.9ms) SELECT COUNT FROM "fact_values" WHERE "fact_values".id IN (199201, 199210, 199211, 199212, 199213, 199214, 199215, 199216, 199217, 199218, 199219, 199220, 199221, 199222, 199223, 199224, 199225, 199226, 199227, 199228, 199229, 200000)
2018-08-22T10:02:54 [l@app][ddca7] Completed 200 OK in 8837ms (Views: 6473.6ms | ActiveRecord: 2280.8ms)
```

Related issues:
Related to Foreman - Bug #21871: [Hammer] PG::Error while non-admin user list... Closed 12/05/2017
Related to Foreman - Bug #25487: 'subtotal' is incorrect in API responses

Closed

Associated revisions

Revision e6503805 - 08/30/2018 07:36 AM - Tomer Brisker
Fixes #24712 - Fix performance regression on API

This commit improves on the fix in 01809199 which has been found to cause significant performance regressions. It also adds several other improvements to the API scope performance, the most significant one being only checking that the scope work in `parent_scope` rather than loading all of it into an in-memory array which can be very heavy.

The original fix was needed because of a bug in the way Rails merges scopes, which will be fixed in Rails 5.2 by https://github.com/rails/rails/pull/29413. Once we upgrade to Rails 5.2 the workaround can be removed since it still comes with worse performance compared to the previous implementation.

History

#1 - 08/24/2018 01:43 PM - Luke Alexander
- Related to Bug #21871: [Hammer] PG::Error while non-admin user listing entities in default location added

#2 - 08/24/2018 01:49 PM - Luke Alexander

Luke Alexander wrote:

Problem:

Foreman API call after upgrade now takes over 8 seconds which previously was under 1 second, I'm not sure if it's isolated to the fact_values endpoint, yet.

Tomer Brisker asked me to link a ref to another ticket which he thinks is the cause of this issue: https://projects.theforeman.org/issues/21871

I noticed a section in the commit: https://github.com/theforeman/foreman/pull/5071/commits/ff3ea42394e9201ec25bef4a41388709509a844f which looked interesting, I reverted that change and then re-ran the curl:

time curl -k -u admin:admin -H "Accept: version=2, application/json" -H "Content-Type: application/json" -X GET "https://theforeman.example.net/api/fact_values?per_page=5&search=name=ipmi_macaddress"

```
{  
"total": 407845,  
"subtotal": 5,  
"page": 1,  
"per_page": 5,  
"search": "name=ipmi_macaddress",  
"sort": {  
  "by": null,  
  "order": null  
},  
"results": {"kr1000.example.net":{"ipmi_macaddress":"00:25:90:0b:24:dc"},"kr110e.example.net":{"ipmi_macaddress":"00:25:90:3f:4c:2c"},"kr1207.example.net":{"ipmi_macaddress":"00:25:90:4d:15:49"},"rb1404.example.net":{"ipmi_macaddress":"00:25:90:52:48:67"},"kr1204.example.net":{"ipmi_macaddress":"00:25:90:6d:f8:cf"}}
```

real 0m0.166s
user 0m0.024s
sys 0m0.004s

03/28/2022
I can confirm this issue. An icinga2 check querying facts takes:
4.67 s
After reverting #21871:
0.59s
I have several checks running, using search and dashboard methods. Only the fact method was affected.

- Found in Releases 1.18.0, 1.18.1, 1.19.0-RC1, 1.19.0-RC2, 1.19.0-RC3 added

- Bugzilla link set to 1622343

- Assignee set to Tomer Brisker
- Status changed from New to Ready For Testing
- Pull request https://github.com/theforeman/foreman/pull/5993 added

Yep, this PR looks good to me, I've applied the diff and the api response is back to similar response time as prior to the upgrade... Thanks for this!

- Fixed in Releases 1.18.2, 1.20.0 added
- Fixed in Releases deleted (1.20.0)
- Related to Bug #25487: 'subtotal' is incorrect in API responses added

Applied in changeset e65038056d56257b32d20b61e2c883fe4b8f3c3a.

- Fixed in Releases 1.18.2, 1.20.0 added
- Fixed in Releases deleted (1.20.0)