Katello - Bug #37346

Fetching Host's details does not scale wrt Hosts Collections

04/11/2024 04:04 PM - Ian Ballou

Status:	Closed		
Priority:	Normal		
Assignee:	Ian Ballou		
Category:	Host Collections		
Target version:	Katello 4.12.1		
Difficulty:		**Fixed in Releases:**	Katello 4.13.0
Triaged:	Yes	**Found in Releases:**	
Bugzilla link:	2249087	**Red Hat JIRA:**	
Pull request:	https://github.com/Katello/katello/pull/10960		

Description

Cloned from https://bugzilla.redhat.com/show_bug.cgi?id=2249087

Description of problem:
Having many Host Collections with many Hosts associated to them, getting a Host details takes tens of seconds.

E.g. having 5k Hosts and 100 Host Collections where five biggest Collections have 4k Hosts each, querying a Host (which is in all the five biggest Collections) takes tens of seconds.

The more Host Collections (and the bigger they are) the Host is assigned to, the longer the query takes.

Version-Release number of selected component (if applicable):
Any version of Katello from the past 9 years

How reproducible:
100%

Steps to Reproduce:
1. Have thousands of Hosts (e.g. the 5k I had). You can populate your Foreman via a very few real Hosts, by repeatedly running on them something like:

```
uuid=$(uuidgen)
echo "{\"dmi.system.uuid\": \"${uuid}\"}" > /etc/rhsm/facts/uuid.facts
hostnamectl set-hostname host-${uuid%%-*}.some.domain.com
subscription-manager clean
subscription-manager register --activationkey ak_test --org RedHat
```

(each iteration of above commands creates one Host)

2. Fetch a Host details few times, like:

```
1. for i in $(seq 1 10); do time curl -ks 'https://localhost/api/v2/hosts/2520' > /dev/null; done 2>&1 | grep real
   real    0m0.269s
   real    0m0.258s
   real    0m0.255s
   real    0m0.253s
   real    0m0.252s
   real    0m0.266s
   real    0m0.667s
   real    0m0.255s
   real    0m0.246s
   real    0m0.256s #
```

3. Create 100 Host Collections, empty so far:

```
collections=100
```

```
( for i in $(seq 1 $collections); do
echo "host-collection create --organization RedHat --name Host_Collection_${i} --unlimited-hosts"
done ) | time hammer shell
```

4. Fetch the Host repeatedly again, to ensure it is still same fast.
5. Add many Hosts to the Host Collections, like e.g.:

```
( for c in $(seq 1 $collections); do
hostids=$(su - postgres -c "psql foreman -c \"COPY (SELECT id FROM hosts WHERE id > $((10*c))) TO STDOUT\"" | tr '\n' ',' | sed "s/,$//g")
echo "host-collection add-host --organization RedHat --name Host_Collection_${c} --host-ids ${hostids}"
done ) | time hammer shell
```

(each Host Collection will differ by 10 Hosts)

6. Fetch the Host repeatedly again; optionally chose different Hosts depending on how many Host Collections they belong to (and how "big" Collections there are).

E.g.:

```
su - postgres -c "psql foreman -c \"SELECT COUNT,host_id FROM katello_host_collection_hosts GROUP BY host_id ORDER BY count DESC LIMIT 5;\""
```

tells you Host IDs that are associated to the most Collections.

Actual results:
2. and 4. shows pretty ow times within one second.
6. shows tens of seconds like:

```
1. for i in $(seq 1 10); do time curl -ks 'https://localhost/api/v2/hosts/2520' > /dev/null; done 2>&1 | grep real
   real    0m24.980s
   real    0m35.901s
   real    0m26.356s
   real    0m23.446s
   real    0m26.306s
   real    0m26.919s
   real    0m36.831s
   real    1m10.623s
   real    0m35.503s
   real    0m19.224s #
```

While a Host in just very few Host Collections is OK:

```
1. for i in $(seq 1 10); do time curl -ks 'https://localhost/api/v2/hosts/51' > /dev/null; done 2>&1 | grep real
   real    0m4.446s
   real    0m1.202s
   real    0m1.704s
   real    0m0.978s
   real    0m1.007s
   real    0m1.190s
   real    0m1.831s
   real    0m0.907s
   real    0m1.049s
   real    0m2.015s #
```

Expected results:
All the times should be within a few seconds.

Additional info:
Enabling psql debugs, here is the source of slowness:

```
2023-11-10T17:28:47 [I|app|c972f5f9] Started GET "/api/v2/hosts/2520" for 127.0.0.1 at 2023-11-10 17:28:47 +0100
2023-11-10T17:28:47 [I|app|c972f5f9] Processing by Api::V2::HostsController#show as JSON
2023-11-10T17:28:47 [I|app|c972f5f9]   Parameters: {"apiv"=>"v2", "id"=>"2520"}
..
2023-11-10T17:28:47 [D|sql|c972f5f9]   Katello::HostCollection Load (0.6ms)  SELECT "katello_host_collections".* FROM "katello_host_collections" INNER JOIN "katello_host_collection_hosts" ON "katello_host_collections"."id" =
"katello_host_collection_hosts"."host_collection_id" WHERE "katello_host_collection_hosts"."host_id" = $1  "host_id", 2520
2023-11-10T17:28:47 [D|sql|c972f5f9]   Host::Managed Load (14.7ms)  SELECT "hosts".* FROM "hosts" INNER JOIN
"katello_host_collection_hosts" ON "hosts"."id" = "katello_host_collection_hosts"."host_id" WHERE "hosts"."type" = $1 AND
"katello_host_collection_hosts"."host_collection_id" = $2  [["type", "Host::Managed"], ["host_collection_id", 201]]
2023-11-10T17:28:47 [D|sql|c972f5f9]   Host::Managed Load (15.1ms)  SELECT "hosts".* FROM "hosts" INNER JOIN
"katello_host_collection_hosts" ON "hosts"."id" = "katello_host_collection_hosts"."host_id" WHERE "hosts"."type" = $1 AND
"katello_host_collection_hosts"."host_collection_id" = $2  [["type", "Host::Managed"], ["host_collection_id", 202]]
..
2023-11-10T17:29:13 [D|sql|c972f5f9]   Host::Managed Load (13.4ms)  SELECT "hosts".* FROM "hosts" INNER JOIN
"katello_host_collection_hosts" ON "hosts"."id" = "katello_host_collection_hosts"."host_id" WHERE "hosts"."type" = $1 AND
"katello_host_collection_hosts"."host_collection_id" = $2  [["type", "Host::Managed"], ["host_collection_id", 300]]
2023-11-10T17:29:13 [D|sql|c972f5f9]   FactValue Load (0.9ms)  SELECT "fact_values".* FROM "fact_values" WHERE
"fact_values"."host_id" = $1  "host_id", 2520
..
2023-11-10T17:29:13 [I|app|c972f5f9] Completed 200 OK in 25707ms (Views: 23842.3ms | ActiveRecord: 1751.7ms | Allocations:
7109377)
```

The queries (for each Host Collection the Host belongs to) "give me all (managed) Hosts from the Host Collection" are the culprit; why do we need them at all..?

Associated revisions

Revision 5457c31c - 04/16/2024 05:28 PM - Ian Ballou

Fixes #37346 - Fetching Host's details does not scale wrt Hosts Collections (#10960)

- Fixes #37346 - Fetching Host's details does not scale wrt Hosts Collections
- Refs #37346 - cache host collection total_hosts

History

#1 - 04/11/2024 04:18 PM - The Foreman Bot

- Status changed from New to Ready For Testing

- Assignee set to Ian Ballou

- Pull request https://github.com/Katello/katello/pull/10960 added

#2 - 04/16/2024 05:28 PM - The Foreman Bot

- Fixed in Releases Katello 4.13.0 added

#3 - 04/16/2024 06:00 PM - Anonymous

- Status changed from Ready For Testing to Closed

Applied in changeset katello|5457c31c5be17842fd5e229c2530bd8c44e6cc39.

#4 - 04/17/2024 05:34 PM - Chris Roberts

- Subject changed from Fetching Host's details does not scale wrt Hosts Collections to Fetching Host's details does not scale wrt Hosts Collections

- Target version set to Katello 4.12.1

- Triaged changed from No to Yes