

Docker Support - Issue #4549

Removing docker manifests from a docker repository takes a long time

03/18/2019 06:47 PM - jsherril@redhat.com

Status: CLOSED - CURRENTRELEASE	Start date:
Priority: Normal	Due date:
Assignee: amacdona@redhat.com	Estimated time: 0:00 hour
Category:	
Sprint/Milestone:	
Severity: 2. Medium	Groomed: No
Version - Docker:	Sprint Candidate: No
Platform Release: 2.21.0	Tags: Pulp 2
Target Release - Docker:	Sprint: Sprint 56
OS:	Quarter:
Triaged: Yes	

Description

Removing all docker manifests from a large docker repo seems to take a long time:

~300 manifests takes ~2 minutes
~2000 manifests takes ~30-40 minutes

Reproducer:

1. Create and sync a docker repo such as: <https://quay.io> datawire/ambassador
2. Remove all docker manifests from the repo: pulp-admin docker repo remove manifest --repo-id=1-docker-dev-7915f7d0-7a98-4131-9c41-1be7b578d442 --not id=foo

Related issues:

Related to Docker Support - Issue #5161: Removing manifest_lists from a repos...	CLOSED - CURRENTRELEASE
Copied to Docker Support - Test #5181: Removing docker manifests from a docke...	CLOSED - COMPLETE

Associated revisions

Revision 76f5894b - 07/30/2019 07:28 PM - amacdona@redhat.com

Flatten queries for content unit removal

<https://pulp.plan.io/issues/4549>

Replace the recursive pattern with a fixed number of larger queries.

Additionally, reorder the content removal to "top down". This will fail more gracefully; failure leaves orphans (safe) rather than user-facing unlinked content (unsafe). This requires the additional plugin step of removing the explicitly given content, which is normally handled by pulp platform.

A side effect of this change is the correction of a bug that did not remove shared content, even if all linked content is removed.

<https://pulp.plan.io/issues/5161>

fixes #5161 fixes #4549

Revision 76f5894b - 07/30/2019 07:28 PM - amacdona@redhat.com

Flatten queries for content unit removal

<https://pulp.plan.io/issues/4549>

Replace the recursive pattern with a fixed number of larger queries.

Additionally, reorder the content removal to "top down". This will fail more gracefully; failure leaves orphans (safe) rather than user-facing unlinked content (unsafe). This requires the additional plugin step of removing the explicitly given content, which is normally handled by pulp platform.

A side effect of this change is the correction of a bug that did not remove shared content, even if all linked content is removed.

<https://pulp.plan.io/issues/5161>

fixes #5161 fixes #4549

Revision ecba1db4 - 08/08/2019 07:25 PM - bherring

Refactor of test_remove.py with changes from 4549

With the refactor of the docker importer's remove function to increase performance, content removal needs to be functional verified.

The cases covered with content post-count verification for all units:

1. Remove all manifest_lists sequentially.
2. Remove all manifests sequentially.
3. Remove all blobs sequentially.
4. Remove all manifest_lists batch.
5. Remove all manifests batch.
6. Remove all blobs batch.
7. Remove some non-shared manifest lists.
8. Remove some non-shared manifest.
9. Remove some shared manifests lists and verify shared units are not recursively removed.
10. Remove some shared manifests and verify shared units are not recursively removed.

The fixture includes:

- * 2 relatively independent manifest lists (no shared manifests, no shared blobs between them)
- * 2 manifest lists that share some (but not all) manifests, and those manifest share some (but not all) blobs. This only requires the creation of 1 manifest list that shares some content with one of the first "independent manifest lists".
- * 2 relatively independent manifests
- * 2 manifests that share (some but not all) blobs

In order to sync the content, each content unit must be recursively related to at least 1 tag.

refs #4549 #5161

closes #5181

Revision fbc90740 - 09/18/2019 08:40 PM - amacdona@redhat.com

Flatten queries for content unit removal

<https://pulp.plan.io/issues/4549>

Replace the recursive pattern with a fixed number of larger queries.

Additionally, reorder the content removal to "top down". This will fail more gracefully; failure leaves orphans (safe) rather than user-facing unlinked content (unsafe). This requires the additional plugin step of removing the explicitly given content, which is normally handled by pulp platform.

A side effect of this change is the correction of a bug that did not remove shared content, even if all linked content is removed.

<https://pulp.plan.io/issues/5161>

fixes #5161 fixes #4549

(cherry picked from commit 76f5894b7c593eafc8498d5215cb7e517cd4624b)

Revision fbc90740 - 09/18/2019 08:40 PM - amacdona@redhat.com

Flatten queries for content unit removal

<https://pulp.plan.io/issues/4549>

Replace the recursive pattern with a fixed number of larger queries.

Additionally, reorder the content removal to "top down". This will fail more gracefully; failure leaves orphans (safe) rather than user-facing unlinked content (unsafe). This requires the additional plugin step of removing the explicitly given content, which is normally handled by pulp platform.

A side effect of this change is the correction of a bug that did not remove shared content, even if all linked content is removed.

<https://pulp.plan.io/issues/5161>

fixes #5161 fixes #4549

(cherry picked from commit 76f5894b7c593eafc8498d5215cb7e517cd4624b)

History

#2 - 03/20/2019 05:00 PM - ipanova@redhat.com

- Triaged changed from No to Yes

- Sprint set to Sprint 50

We need to investigate this.

#3 - 03/27/2019 09:05 PM - dkliban@redhat.com

- Status changed from NEW to ASSIGNED

- Assignee set to dkliban@redhat.com

#4 - 03/29/2019 10:01 PM - rchan

- Sprint changed from Sprint 50 to Sprint 51

#5 - 04/03/2019 03:02 PM - dkliban@redhat.com

- Status changed from ASSIGNED to NEW

- Assignee deleted (dkliban@redhat.com)

#6 - 04/15/2019 09:59 PM - bmbouter

- Tags Pulp 2 added

#7 - 04/23/2019 09:08 PM - daviddavis

- Sprint changed from Sprint 51 to Sprint 52

#8 - 05/13/2019 03:06 PM - rchan

- Sprint changed from Sprint 52 to Sprint 53

#9 - 05/31/2019 03:27 PM - dkliban@redhat.com

- Sprint deleted (Sprint 53)

#10 - 07/01/2019 02:48 PM - dkliban@redhat.com

- Sprint set to Sprint 55

#11 - 07/09/2019 04:39 AM - amacdonga@redhat.com

- Status changed from NEW to ASSIGNED

- Assignee set to amacdonga@redhat.com

#12 - 07/10/2019 12:09 AM - amacdonga@redhat.com

- File almost-all-man-lists-busybox added

I was able to reproduce this with the busybox repository (I'll make a bigger vm to do this again with datawire/ambassador next week) and the slowdown is already apparent, taking about 2 minutes to remove all the manifest-lists.

I've attached the c_profile of that task. How to view: <https://docs.pulpproject.org/dev-guide/debugging.html#analyzing-profiles>

Potential workaround:

The queries for removing content are much more strenuous than adding content. A potential workaround could be to create a new repository and add only the manifest lists that should remain. If for some reason we can't work out a big improvement, I'll profile the workaround to compare.

#13 - 07/12/2019 03:27 PM - dkliban@redhat.com

- Sprint changed from Sprint 55 to Sprint 56

#14 - 07/20/2019 03:24 PM - amacdona@redhat.com

- Related to Issue #5161: Removing manifest_lists from a repository does not purge all newly unlinked manifests added

#15 - 07/20/2019 03:31 PM - amacdona@redhat.com

There is a minor bug in current Pulp2 removal which will be fixed by this change.

<https://pulp.plan.io/issues/5161>

Because of that bug, this fix will include a very minor behavior change. More manifests and blobs will be recursively removed when removing manifest_lists that share manifests. The manifests and blobs that will now be removed would have been inaccessible by tags, so should not affect users.

#16 - 07/24/2019 02:58 PM - bherring

- Copied to Test #5181: Removing docker manifests from a docker repository takes a long time added

#17 - 07/29/2019 05:45 PM - amacdona@redhat.com

Time improvement

Removal of 210 Manifests (unpatched): 84 minutes

Removal of 210 Manifests (patched): 45 seconds

Removal of 2997 Manifests(patch): 1 minute 19 seconds

Removal of 4381 Manifests (patched): 1 minute 30 seconds

Memory Usage

This speed improvement comes as a tradeoff for memory usage. Unpatched, the memory usage of the celery process was fixed and did not increase with the size of the call. (This is because pulp worked through the task one content unit at a time.)

With the patch, memory usage will increase with the number of content units specified in the removal call.

Patched removal peak RES memory usage of celery process:

REMOVAL : PEAK RES (k)

30 manifests: 234444

145 manifests: 251908

210 manifests: 260616

2996 manifests: 792260

4381 manifests: 902488

(Note: these numbers can vary based on the specific content. Manifests that reference a higher number of blobs will use more memory, etc.)

Reference tasks: (high memory tasks with docker):

Sync datawire/ambassador (2926 tags): 317572

Resting celery: 75260

COPY 2926 tags: 443820

To reproduce:

Big repos from quay.io: datawire/ambassador, calico/typha

Start (or restart) 1 pulp worker between each task

Use `grep VmHWM /proc/\$pid/status` to determine peak RES usage

Conclusion:

Because memory usage scales up with removal size (limited only by repo size), there is a theoretical memory problem for arbitrarily large docker repositories. However even using a very large repo (~50,000 content units), the peak RES memory was below 1 GB. Therefore, my best guess is that this problem will not be a practical concern for real docker repositories in the wild.

#18 - 07/29/2019 07:52 PM - amacdona@redhat.com

- Status changed from ASSIGNED to POST

https://github.com/pulp/pulp_docker/pull/385/

#19 - 07/29/2019 10:26 PM - mmccune@redhat.com

nice work on the optimization. IMO, the memory consumption for this call is a worthy tradeoff for the performance increases we are getting with your optimization. If you were quoting numbers in the 2-10G consumption I'd be concerned.

#20 - 08/01/2019 07:11 PM - amacdona@redhat.com

- Status changed from POST to MODIFIED

Applied in changeset [pulp_docker|76f5894b7c593eafc8498d5215cb7e517cd4624b](#).

#21 - 08/18/2019 07:02 AM - amacdona@redhat.com

Applied in changeset [76f5894b7c593eafc8498d5215cb7e517cd4624b](#).

#22 - 08/19/2019 11:30 PM - dalley

- Platform Release set to 2.21.0

#23 - 09/18/2019 08:42 PM - amacdona@redhat.com

Applied in changeset [pulp_docker|fbc9074078bf4c324c3820bd3201449410ba95ad](#).

#24 - 09/25/2019 10:46 PM - dalley

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE

Files

almost-all-man-lists-busybox	210 KB	07/09/2019	amacdona@redhat.com
------------------------------	--------	------------	---------------------