Arvados - Bug #12990

[FUSE] Access shared/ is inefficient

01/23/2018 02:31 PM - Peter Amstutz

Status:	In Progress	**Start date:**	
Priority:	Normal	**Due date:**	
Assigned To:	Peter Amstutz	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

The shared/ directory of FUSE has several issues:

1. no update lock, may start overlapping updates in separate threads
2. no incremental lookup of individual names, always loads full list, bad for scaling
3. fetches full record, which may include description or properties payload which is not used by wastes bandwith

Related issues:

Related to Arvados - Story #13146: [API] Endpoint to get projects shared with me	**Resolved**	**08/15/2018**

Associated revisions

Revision ae35f42f - 01/23/2018 05:40 PM - Peter Amstutz

Merge branch '12990-fuse-shared' refs #12990

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

History

#1 - 01/23/2018 02:31 PM - Peter Amstutz

- Status changed from New to In Progress

#2 - 01/23/2018 02:37 PM - Peter Amstutz

- Description updated

#3 - 01/23/2018 03:59 PM - Peter Amstutz

12990-fuse-shared @ 0dcf9daff8fce376f20f125c3ef867333976c18c

Addresses points (1) and (3) but not incremental lookup (this turns out to be hard due to the way the contents of shared/ is determined).

#4 - 01/23/2018 04:34 PM - Tom Clegg

LGTM

This looks like it should fix the "flood the apiserver with many threads of groups#list requests" issue we're seeing.

I'm not certain, but I see a couple of other issues that (if they're real) are probably worth fixing:

- ProjectDirectory and SharedDirectory don't seem to call fresh() after updating, like CollectionDirectory does. Does this mean once they go stale, they stay stale forever, and every lookup triggers a refresh?
- If N threads decide that self is stale, they all line up for updating_lock, and do their updates serially. But the first one should (according to the previous point, at least) set fresh(), which means the next N-1 threads will dutifully do their laborious updates even though self is already fresher than they could possibly have wanted it to be back when they decided to update. Perhaps it would be better to do one of
  - Check stale() after acquiring _updating_lock, so the last N-1 threads just wait for the update that's already in progress to finish, and don't bother doing their own.
  - Use acquire(false) to do a non-blocking lock. This is a bit different in that it knowingly returns stale results, but in the case of SharedDirectory maybe this kind of race is OK, since we generally only detect staleness using a race-prone timer anyway?

#5 - 01/23/2018 04:43 PM - Tom Clegg

Tom Clegg wrote:

> I'm not certain, but I see a couple of other issues that (if they're real) are probably worth fixing:

(from irc) Not real. merge() sets fresh flag. "Check stale() after acquiring" already happens.

#6 - 01/23/2018 05:08 PM - Peter Amstutz

Passed tests here https://ci.curoverse.com/job/developer-run-tests-services-fuse/564/

#7 - 01/23/2018 07:59 PM - Tom Morris

- Assigned To set to Peter Amstutz

#8 - 01/23/2018 08:38 PM - Peter Amstutz

To do this more efficient likely requires a new API endpoint. The way arv-mount currently determines what to list in "shared" currently requires looking at all projects and finding the ones where owner_uuid is not another project which is visible to us (meaning: users, non-project groups, or shared subprojects where the parent is not visible). This is expensive to compute on the client, but can probably be accomplished with a single query on the API server.

#9 - 02/01/2018 08:30 PM - Peter Amstutz

- Target version changed from 2018-01-31 Sprint to Arvados Future Sprints

#10 - 02/28/2018 09:08 PM - Peter Amstutz

- Related to Story #13146: [API] Endpoint to get projects shared with me added

#11 - 02/28/2018 09:08 PM - Peter Amstutz

Discussion about API endpoint moved to #13146

#12 - 07/06/2021 09:22 PM - Peter Amstutz

- Target version deleted (Arvados Future Sprints)