Arvados - Bug #10584

[FUSE] high memory consumption (possible leak) in long-running arv-mount

11/22/2016 02:43 PM - Joshua Randall

Status:	New	**Start date:**	11/22/2016
Priority:	Normal	**Due date:**	
Assigned To:		**% Done:**	50%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

We have a (little used) arv-mount that has been running since 6th September.

It was started with the command line:
`/usr/bin/python2.7 /usr/bin/arv-mount /tmp/keep_jr17`

Since no `--file-cache` or `--directory-cache` options were given, those should have been the defaults of 256MiB and 128MiB. If I start a new arv-cache also with defaults and then read some large data through it and exercise some large directories (such as doing a find in `by_tag`), I am able to get memory usage up to 514MB, which seems reasonable.

However, the arv-mount that has been running for the past 77 days is now taking up 15GB of RAM!

I suspect this issue **might** be related to the increasing memory usage I observed and reported in #10535 when the python SDK test suite got stuck in a tight PollClient loop forever (where "forever" is until it ran the system out of memory).

Subtasks:

Task # 11890: Review 10584-fuse-stop-threads	**Resolved**
Task # 12067: Instrumentation	**New**

Associated revisions

Revision 1e31815d - 07/07/2017 03:27 PM - Peter Amstutz

Merge branch '10584-fuse-stop-threads' refs #10584

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <peter.amstutz@curoverse.com>

History

#1 - 03/01/2017 05:29 PM - Tom Morris

- Subject changed from high memory consumption (possible leak) in long-running arv-mount to [FUSE] high memory consumption (possible leak) in long-running arv-mount

- Target version set to Arvados Future Sprints

#2 - 06/21/2017 07:29 PM - Tom Morris

- Target version changed from Arvados Future Sprints to 2017-07-05 sprint

#3 - 06/21/2017 07:30 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

#4 - 07/03/2017 01:42 PM - Peter Amstutz

Some theories:

- This might be related/due to https://dev.arvados.org/issues/11158 → it is that it is trying to enumerate the entire home directory and it uses up all memory trying to store the full contents.
- Cache management clears releases unused Collection objects. However, those Collection objects may have prefetch threads. If they don't get stopped, they will leak. *

#5 - 07/05/2017 06:25 PM - Peter Amstutz

- Target version changed from 2017-07-05 sprint to 2017-07-19 sprint

#6 - 07/06/2017 02:17 AM - Peter Amstutz

Ensure get/put threads are stopped before releasing reference to Collection object. Unclear if this is the source of the problem, but seems like a good idea regardless.

#7 - 07/06/2017 07:08 PM - Lucas Di Pentima

The thread stopping code was added on a CollectionDirectoryBase subclass, is it possible for this problem to happen with TmpCollectionDirectory objects too? Maybe it's better to do the thread stopping on CollectionDirectoryBase?

#8 - 07/07/2017 12:55 PM - Peter Amstutz

Lucas Di Pentima wrote:

> The thread stopping code was added on a CollectionDirectoryBase subclass, is it possible for this problem to happen with TmpCollectionDirectory objects too? Maybe it's better to do the thread stopping on CollectionDirectoryBase?

CollectionDirectoryBase objects are used to hold Subcollection objects, which don't have a stop_threads() method.

TmpCollectionDirectory are not candidates for cache eviction (persisted() is False). The finalize() method already calls stop_threads().

The difference between clear() and finalize() is that clear() is called when we want to evict an inode's cached contents, whereas finalize() is called when the inode will be deleted entirely.

#9 - 07/07/2017 02:47 PM - Lucas Di Pentima

Ok, so this looks good to me. Thanks!

#10 - 07/19/2017 06:44 PM - Tom Morris

- Target version changed from 2017-07-19 sprint to 2017-08-02 sprint

#12 - 08/02/2017 06:55 PM - Peter Amstutz

- Look at user interaction history with keep
- Track metrics
- Instrumentation to report memory usage / ownership

#13 - 08/02/2017 06:57 PM - Peter Amstutz

- Target version changed from 2017-08-02 sprint to 2017-08-16 sprint

#14 - 08/16/2017 07:08 PM - Tom Morris

- Target version changed from 2017-08-16 sprint to 2017-08-30 Sprint

#15 - 08/30/2017 06:50 PM - Peter Amstutz

- Assigned To deleted (Peter Amstutz)

- Target version changed from 2017-08-30 Sprint to 2017-09-13 Sprint

#16 - 08/30/2017 07:47 PM - Tom Morris

- Target version changed from 2017-09-13 Sprint to Arvados Future Sprints

#17 - 08/30/2017 07:50 PM - Tom Clegg

Might be worth running the "retry PUT" test many times in a row. At least once I've seen the test suite get stuck there using lots of memory.

#18 - 07/06/2021 09:21 PM - Peter Amstutz

- Target version deleted (Arvados Future Sprints)