

Arvados - Bug #16480

[keep-balance] should not timeout/fail when keepstore index takes more than 5 minutes

06/02/2020 08:46 PM - Tom Clegg

Status:	Resolved	Start date:	06/16/2020
Priority:	Normal	Due date:	
Assigned To:	Tom Clegg	% Done:	100%
Category:	Keep	Estimated time:	0.00 hour
Target version:	2020-07-01 Sprint		
Description			
<p>Currently sdk/go/arvados.Client uses a 5-minute request timeout when the caller (like keep-balance) doesn't provide a custom *http.Client. This isn't enough time to complete a keepstore index request on a cloud install with lots of data. When the timeout expires, keep-balance just gives up on the current iteration, goes to sleep, and waits for the next one, which is unlikely to fare any better.</p> <p>It may be worth having an idle timer to detect hung connections, but aside from that keep-balance should wait as long as it takes for keepstore to return an index response.</p>			
Subtasks:			
Task # 16486: Review 16480-keep-balance-index-timeout			Resolved

Associated revisions

Revision 405b13d5 - 06/22/2020 03:36 PM - Tom Clegg

Merge branch '16480-keep-balance-index-timeout'

fixes #16480

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@tomclegg.ca>

History

#2 - 06/03/2020 03:59 PM - Tom Clegg

- Target version set to 2020-06-17 Sprint

- Assigned To set to Tom Clegg

#4 - 06/15/2020 08:51 PM - Tom Clegg

- Status changed from New to In Progress

#5 - 06/16/2020 02:12 PM - Tom Clegg

16480-keep-balance-index-timeout @ [2bc1a7a89597ab02aaef84b82fdc51f8e375b79](https://ci.arvados.org/view/Developer/job/developer-run-tests/1919/) --
<https://ci.arvados.org/view/Developer/job/developer-run-tests/1919/>

This changes the timeout to 24h. I'm inclined to punt on the "idle timer to detect hung connections" part.

#6 - 06/16/2020 07:54 PM - Ward Vandewege

Tom Clegg wrote:

16480-keep-balance-index-timeout @ [2bc1a7a89597ab02aaef84b82fdc51f8e375b79](https://ci.arvados.org/view/Developer/job/developer-run-tests/1919/) --
<https://ci.arvados.org/view/Developer/job/developer-run-tests/1919/>

This changes the timeout to 24h. I'm inclined to punt on the "idle timer to detect hung connections" part.

OK to punt on the "idle timer", that's a nice to have. But maybe shorten the timeout a bit, 24 hours seems a bit excessive?

Review comment:

Removing the 5 minute timeout from the definition of *DefaultSecureClient* and *InsecureHTTPClient* means that the *setup()* function in *lib/controller/handler.go* now creates a *secureClient* and *insecureClient* without timeout. Is that intentional? Since *DefaultSecureClient* and *InsecureHTTPClient* are exported, shouldn't there be a default timeout on them?

Otherwise, LGTM, thanks.

#7 - 06/17/2020 03:32 PM - Tom Clegg

- Target version changed from 2020-06-17 Sprint to 2020-07-01 Sprint

#8 - 06/19/2020 02:58 PM - Tom Clegg

Ward Vandewege wrote:

OK to punt on the "idle timer", that's a nice to have. But maybe shorten the timeout a bit, 24 hours seems a bit excessive?

I suppose the only way to avoid choosing a bad timeout is to make it configurable so the admin can choose a bad timeout themselves, so I made a configurable timeout for the entire keep-balance operation, defaulting to 6h. Is that a good compromise between annoyingly-long-for-small-sites and annoyingly-short-for-big-sites?

Removing the 5 minute timeout from the definition of *DefaultSecureClient* and *InsecureHTTPClient* means that the *setup()* function in *lib/controller/handler.go* now creates a *secureClient* and *insecureClient* without timeout. Is that intentional? Since *DefaultSecureClient* and *InsecureHTTPClient* are exported, shouldn't there be a default timeout on them?

Controller already uses a context deadline on incoming requests to enforce *API.RequestTimeout*, and propagates that context to outgoing proxy requests. So we were effectively limiting outgoing reqs *min(5m, RequestTimeout)* -- using only *RequestTimeout* seems better.

Generally, I'm thinking it's OK to change the exported default clients' timeouts to zero -- that's how *http.DefaultClient* does it, probably a hint that we should let the caller provide a timeout/context when needed.

16480-keep-balance-index-timeout @ [fd080b34a321cbd6593d69f427b9eaeab890712f](https://ci.arvados.org/view/Developer/job/developer-run-tests/1925/) --
<https://ci.arvados.org/view/Developer/job/developer-run-tests/1925/>

#9 - 06/19/2020 09:48 PM - Ward Vandewege

Tom Clegg wrote:

Ward Vandewege wrote:

OK to punt on the "idle timer", that's a nice to have. But maybe shorten the timeout a bit, 24 hours seems a bit excessive?

I suppose the only way to avoid choosing a bad timeout is to make it configurable so the admin can choose a bad timeout themselves, so I made a configurable timeout for the entire keep-balance operation, defaulting to 6h. Is that a good compromise between annoyingly-long-for-small-sites and annoyingly-short-for-big-sites?

Removing the 5 minute timeout from the definition of *DefaultSecureClient* and *InsecureHTTPClient* means that the *setup()* function in *lib/controller/handler.go* now creates a *secureClient* and *insecureClient* without timeout. Is that intentional? Since *DefaultSecureClient* and *InsecureHTTPClient* are exported, shouldn't there be a default timeout on them?

Controller already uses a context deadline on incoming requests to enforce *API.RequestTimeout*, and propagates that context to outgoing proxy requests. So we were effectively limiting outgoing reqs *min(5m, RequestTimeout)* -- using only *RequestTimeout* seems better.

Generally, I'm thinking it's OK to change the exported default clients' timeouts to zero -- that's how *http.DefaultClient* does it, probably a hint that we should let the caller provide a timeout/context when needed.

16480-keep-balance-index-timeout @ [fd080b34a321cbd6593d69f427b9eaeab890712f](https://ci.arvados.org/view/Developer/job/developer-run-tests/1925/) --
<https://ci.arvados.org/view/Developer/job/developer-run-tests/1925/>

Cool, this LGTM.

#10 - 06/22/2020 04:09 PM - Anonymous

- % Done changed from 0 to 100

- Status changed from In Progress to Resolved

Applied in changeset [arvados|405b13d50e203958968427a2642bc18026a0c227](https://ci.arvados.org/view/Developer/job/developer-run-tests/1925/).

#11 - 10/07/2020 02:11 AM - Peter Amstutz

- Release set to 25