

Arvados - Bug #7253

[Keep] [Data Manager] [SDKs] Test behavior in the face of API and manifest errors

09/10/2015 02:46 AM - Tom Clegg

Status:	Resolved	Start date:	09/10/2015
Priority:	Normal	Due date:	
Assigned To:	Radhika Chippada	% Done:	100%
Category:	Keep	Estimated time:	0.00 hour
Target version:	2015-12-02 sprint		
Description			
Tests must ensure Data Manager stops without deleting anything in all of the following situations:			
<ul style="list-style-type: none">• A network error is encountered when getting the collection list from the API server.• An API error is encountered when getting the collection list (for example, its token expires between fetching page N and fetching page N+1).• Its token does not allow it to see all collections (i.e., it is not an admin token). (test already added in #6260, but worth double-checking!)• A collection manifest does not have the expected format. (In particular, the current code stops parsing a stream as soon as it sees a token that doesn't satisfy its locator regexp. This approach is fragile: it can cause data loss if a manifest contains a block locator that is acceptable to the API server's validation method but doesn't match Data Manager's regexp. We should assume the manifest and locator formats will change over time, and make sure a too-old version of Data Manager does not delete valuable data when asked to do garbage collection for a new/upgraded Arvados installation.)			
Subtasks:			
Task # 7785: Review branch: 7253-datamanager-test-errors in arvados and 7253-add-go-sdk...			Resolved
Related issues:			
Related to Arvados - Story #6260: [Keep] Integration test between data manage...			Resolved 08/19/2015

Associated revisions

Revision 24129230 - 11/26/2015 03:45 PM - Radhika Chippada

refs #7253
Merge branch '7253-add-go-sdk-manifest'

Revision 24129230 - 11/26/2015 03:45 PM - Radhika Chippada

refs #7253
Merge branch '7253-add-go-sdk-manifest'

Revision 72189590 - 11/26/2015 03:46 PM - Radhika Chippada

closes #7253
Merge branch '7253-datamanager-test-errors'

Revision 719dae9a - 11/26/2015 06:10 PM - Radhika Chippada

refs #7253
Merge branch '7253-datamanager-test-errors'

History

#1 - 09/16/2015 03:29 PM - Brett Smith

- Target version set to Arvados Future Sprints

#2 - 11/12/2015 02:09 AM - Brett Smith

- Assigned To set to Radhika Chippada
- Target version changed from Arvados Future Sprints to 2015-12-02 sprint

#3 - 11/19/2015 03:55 PM - Radhika Chippada

- Status changed from New to In Progress

#4 - 11/25/2015 07:23 PM - Radhika Chippada

sdk/go/manifest/manifest.go -> BlockIterWithDuplicates and parseManifestStream are updated to look for errors in manifest_text and return them. Made updates in datamanager/collection to look for these errors.

Several tests to verify API errors and manifest errors during GetCollections have been added. The specific test "An API error is encountered when getting the collection list (for example, its token expires between fetching page N and fetching page N+1)" has not been added due to the difficulty in simulating such as error, but instead other API error simulation tests were added.

#5 - 11/25/2015 09:14 PM - Tom Clegg

sdk/go/arvadostest/testing.go:

testing.go -> stub.go

StatusAndBody { ResponseStatus, ResponseBody } -> StubResponse { Status, Body } ?

In APIStub: Data -> Responses

KeepServerStub seems to be identical to APIStub; if so, just say type KeepServerStub APIStub instead of having two copies of the code. (Apparently I should have noticed this in #7490...)

sdk/go/manifest/manifest.go:

Can we stop using "m" for both "Manifest" and "ManifestStream" -- maybe "ms" for ManifestStream? At least we should fix BlockIterWithDuplicates where m starts out being a *Manifest but is then reused to hold a ManifestStream in the goroutine loop.

in parseManifestStream:

```

• █ for j := range m.FileTokens {
-   _ , _ , _ , err := parseFileToken(fileTokens[j])
+   for _, ft := range m.FileTokens {
+       _ , _ , _ , err := parseFileToken(ft)

```

Instead of wrapping BlockLocator in ManifestBlockLocator, how about:

- Specify in the BlockIterWithDuplicates comment that the caller must check m.Err after the returned channel closes, in order to detect parse errors.
- Set m.Err = err (and don't send anything to blockChannel) if blockdigest.ParseBlockLocator() fails.

This seems to make a copy of b instead of just using b... but I'm not sure why?

```

• █ expectBlockLocator(t, b, ...
+   expectBlockLocator(t, blockdigest.BlockLocator{b.Digest, b.Size, b.Hints}, ...

```

sdk/go/manifest/manifest_test.go:

Instead of having one "success" case in TestBlockIterWithBadManifest, just test bad manifests -- and check manifest.Err == nil in all the other test cases that test BlockIterWithDuplicates.

Test: ". acbd18db4cc2f85cedef654fcc4a4d8+3 0:1:file1.txt. acbd18db4cc2f85cedef654fcc4a4d8+3 1:2:file3.txt"

Test: ". acbd18db4cc2f85cedef654fcc4a4d8+3 0:1:file1.txt\n. acbd18db4cc2f85cedef654fcc4a4d8+3 :file2.txt\n"

Test: ". acbd18db4cc2f85cedef654fcc4a4d8+3 acbd18db4cc2f85cedef654fcc4a4d8+3\n"

Consume the whole channel, not just the first block:

```

• █

```

Tests would be a bit less repetitive like this:

```

-   testData := APITestData{}
-   testData.data = dataMap
-   testData.expectedError = "Invalid stream name: badstreamname"
-
+   testGetCollectionsAndSummarize(c, testData)
+   testGetCollectionsAndSummarize(c, APITestData{
+       data:          dataMap,
+       expectedError: "Invalid stream name: badstreamname",
+   })

```

#6 - 11/26/2015 05:34 AM - Radhika Chippada

Updated to address all those comments. Just a couple things:

- Good catch about the APIStub and KeepServerStub doing the same thing. After all the cleanup and all, it became obvious. I now have only one and called it ServerStub
- Regarding "This seems to make a copy of b instead of just using b... but I'm not sure why? expectBlockLocator(t, b, ..." Here "b" is manifest.BlockLocator, whereas expectBlockLocator expects a blockdigest.BlockLocator

Thanks.

#7 - 11/26/2015 03:37 PM - Tom Clegg

The second line here is unnecessary (in testGetCollectionsAndSummarize): results can't be nil because it's not a pointer, and even if it was a pointer, the first line would have already crashed if it was nil...

```
c.Assert(results.Err, IsNil)
c.Assert(results, NotNil)
```

Everything else LGTM, thanks.

#8 - 11/26/2015 03:48 PM - Tom Clegg

LGTM @ [cbb8375](#)

#9 - 11/26/2015 04:00 PM - Radhika Chippada

- Status changed from *In Progress* to *Resolved*

- % Done changed from 0 to 100

Applied in changeset arvados|commit:72189590b0fc8555afaf54d7a27eb609b2e157d0.