Arvados - Bug #5319

[API] Portable data hash mismatches for collections with location hints (+K)

02/26/2015 04:31 PM - Bryan Cosca

Status:	Resolved	**Start date:**	03/03/2015
Priority:	Normal	**Due date:**	
Assigned To:	Brett Smith	**% Done:**	100%
Category:	API	**Estimated time:**	0.00 hour
Target version:	2015-03-11 sprint		

Description

portabledatahash.png

Subtasks:

Task # 5369: Review 5319-fix-hinted-collection-hashes-wip	**Resolved**

Related issues:

Related to Arvados - Support #5251: [Support] Fix bugs and write tests (first...	**Resolved**

Associated revisions

Revision 19656a41 - 03/09/2015 03:35 PM - Brett Smith

Merge branch '5319-fix-hinted-collection-hashes-wip'

Closes #5319, #5369.

Revision efd59dbd - 03/09/2015 04:12 PM - Brett Smith

5319: Include CurrentApiClient in migration model.

Needed to use system_user_uuid. Refs #5319.

Revision 9877c891 - 03/16/2015 04:41 PM - Brett Smith

Merge branch '5319-collection-pdh-fix-performance-wip'

Refs #5319.

Revision 7e5416c8 - 03/20/2015 06:08 PM - Brett Smith

Merge branch '5319-collection-pdh-fix-performance-2-wip'

Refs #5319.

Revision a307f4b9 - 03/20/2015 09:16 PM - Brett Smith

5319: Bound search in collection PDH fix migration.

Only search collections that exist when the migration begins. This
helps avoid an infinite loop where we yield a bad collection, it gets
fixed a new replacement is created, and then we find and yield the
replacement… Refs #5319.

History

#1 - 02/26/2015 04:33 PM - Bryan Cosca

- File portablehash-2.png added

My current workaround is to select all the files and then create a new collection, which i guess makes a new hash. and I can use that new collection.

portablehash-2.png

#2 - 02/26/2015 06:18 PM - Brett Smith

- Subject changed from Portable Data hash does not match computed hash to [Workbench] Can't copy collection between projects due to portable data hash mismatch

- Category set to Workbench

- Target version set to Bug Triage

#3 - 02/26/2015 06:29 PM - Brett Smith

- Status changed from New to In Progress

- Assigned To set to Brett Smith

#4 - 02/26/2015 06:29 PM - Brett Smith

- Target version changed from Bug Triage to 2015-03-11 sprint

#5 - 02/26/2015 06:40 PM - Bryan Cosca

qr1hi-4zz18-yrocuypi31jo3ru and qr1hi-4zz18-h03wet37z45z2c9 are known culprits

#6 - 02/26/2015 07:21 PM - Brett Smith

The affected collections have manifests with Keep location hints in them (+Kqr1hi), and the portable data hash corresponds to a manifest text including those hints. Now that API server disregards all hints when calculating a portable data hash, a straightforward copy fails.

#7 - 02/26/2015 10:52 PM - Brett Smith

I believe this change is our culprit, specifically on old line 87/new line 106. That's validation-related code that changed from using self.manifest_text to portable_manifest_text. portable_manifest_text strips all hints, so that would explain the difference.

Questions:

- What's the right policy here? Should the PDH correspond to the text without any hints, or just without access tokens? Tom made this change, so I'm guessing it's the former…
- Once we figure that out, do we need a migration to correct the PDHs of older collections?

#8 - 02/27/2015 08:05 PM - Brett Smith

- Subject changed from [Workbench] Can't copy collection between projects due to portable data hash mismatch to [API] Portable data hash mismatches for collections with location hints (+K)

- Category changed from Workbench to API

#9 - 03/02/2015 03:07 PM - Peter Amstutz

+1 to doing a migration. It should probably include updating the PDH of the collection in pipelines templates.

#10 - 03/02/2015 07:16 PM - Tom Clegg

Yes, PDH should stay the same when the hints change (e.g., when the data comes and goes from different storage systems, and when the manifest is copied between instances).

As for cleaning up the existing incorrect PDH values:

A migration that *changes* the PDH values would make the old/broken PDH unfindable, which means either editing evidence about existing jobs and pipeline instances (yuck) or making them un-repeatable (also yuck, although less so). Perhaps we can make copies of the affected collections instead, so we use the new PDH for future work but the old ones can still be looked up. The remaining challenge would be to prevent the old/bogus copies from showing up as confusing duplicates in searches.

(A "copy" API call would help address this (and, perhaps more importantly, #5096), but seems like excessive creep to do here, and doesn't address the need for a migration to fix the bogus PDH values.)

#11 - 03/02/2015 08:12 PM - Brett Smith

Tom Clegg wrote:

> A migration that *changes* the PDH values would make the old/broken PDH unfindable, which means either editing evidence about existing jobs and pipeline instances (yuck) or making them un-repeatable (also yuck, although less so). Perhaps we can make copies of the affected collections instead, so we use the new PDH for future work but the old ones can still be looked up. The remaining challenge would be to prevent the old/bogus copies from showing up as confusing duplicates in searches.

The longer-term plan is for Workbench to ignore objects that expire in the future (showing them only in a Trash), right? If that's the case, what if the migration set expires_at for the affected Collections far in the future, like 2038? That way they'd still be easy to find for jobs, and we have a way to hide them in Workbench that's consistent with other plans.

#12 - 03/02/2015 10:11 PM - Tom Clegg

Brett Smith wrote:

> The longer-term plan is for Workbench to ignore objects that expire in the future (showing them only in a Trash), right? If that's the case, what if the migration set expires_at for the affected Collections far in the future, like 2038? That way they'd still be easy to find for jobs, and we have a way to hide them in Workbench that's consistent with other plans.

Yes, that sounds like it should work nicely.

#13 - 03/05/2015 02:38 PM - Peter Amstutz

General comment: my understanding is that best practices for migrations to modify the tables directly using SQL, instead of using models, since the model code can change to no longer match the assumptions at the time the migration was written. However, since most of the time the migrations are run promptly, this hasn't been much of an issue in practice.

- Since existing Collection records are getting updated, perhaps it should be writing updates to the log table?
- Why does each_bad_collection search for tags matching \+[B-Z] and not \+[A-Z]?
- Have you tried this out on a test database?

#14 - 03/06/2015 04:37 PM - Brett Smith

Peter Amstutz wrote:

> General comment: my understanding is that best practices for migrations to modify the tables directly using SQL, instead of using models, since the model code can change to no longer match the assumptions at the time the migration was written. However, since most of the time the migrations are run promptly, this hasn't been much of an issue in practice.

I have reworked the migration to follow the Rails Guide's advice about this.

> - Since existing Collection records are getting updated, perhaps it should be writing updates to the log table?

Sure, added.

> - Why does each_bad_collection search for tags matching \+[B-Z] and not \+[A-Z]?

The API server has been pretty good for a while now about making sure that manifests don't get saved with permission hints. But it can't hurt to be sure, so changed.

> - Have you tried this out on a test database?

Yes. The easiest way I've found is to add this test collection fixture:

```
with_location_hints:
  owner_uuid: zzzzz-tpzed-xurymjxw79nv3jz
  created_at: 2014-02-03 17:22:54.000000000 Z
  modified_by_client_uuid: zzzzz-ozdt8-brczlopd8u8d0jr
  modified_by_user_uuid: zzzzz-tpzed-d9tiejq69daie8f
  modified_at: 2014-02-03 17:22:54.000000000 Z
  portable_data_hash: 3eab909704896fd5e1c70cd135ab6b8d+52
  updated_at: 2014-02-03 17:22:54.000000000 Z
  uuid: bcsdv-4zz18-6pxgxibl4vmyt09
  manifest_text: |
    . 37b51d194a7513e45b56f6524f2d51f2+3+Kfhqwh 0:3:bar
  name: manifest with location hints
  description:
  properties: {}
```

With that in place, you can db:fixtures:load, then db:migrate and db:rollback and observe the results.

Now at cea7dc4. Thanks.

#15 - 03/09/2015 01:12 PM - Peter Amstutz

I noticed that it doesn't actually update the contents of manifest_text. Is that intentional or an oversight?

modified_at also isn't updated, and the copied collection has created_at copied instead of set to now. This might be the right thing to do, but I'd like to hear your rationale.

Everything else looks good.

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

Peter Amstutz wrote:

> I noticed that it doesn't actually update the contents of manifest_text. Is that intentional or an oversight?

Intentional: that's what the API server is currently doing. See strip_manifest_text in the Collection model: the only hints it strips are signatures. You can also see qr1hi-4zz18-cs8ezglaomoym37, which is the collection created by creating a "new" collection with all the same files as the original: its manifest text still has the hints. And this makes sense to me. This way, the portable data hash stays the same as long as the data does, but we retain hint information as much as possible to help find that data faster.

> modified_at also isn't updated, and the copied collection has created_at copied instead of set to now. This might be the right thing to do, but I'd like to hear your rationale.

I guess the question boils down to, should those timestamps be true to the literal database record, or should they be true to the underlying collection? From the database perspective, I think your ideas are the right ones. I was thinking about it more at the collection level: the *collection* isn't new or changed, we've just patched up how we address it in the database. And so the logic reflects that.

To the extent that this is a bugfix migration that we're trying to keep invisible to the user (and admittedly, that's still not very invisible), I think this approach supports that goal. Imagine a user doing API requests for collections ordered by one of these timestamps descending. I think it would surprise them to see "old" collections suddenly spring up to the top of those results. We can't hide the second record from them, but I think this approach puts it in a context that can make it easier to see where it's coming from.

Thanks.

#17 - 03/09/2015 03:31 PM - Peter Amstutz

Thanks. LGTM

#18 - 03/09/2015 03:40 PM - Brett Smith

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

Applied in changeset arvados|commit:19656a41f019488120f950b06ecf9e19074b11a3.

#19 - 03/11/2015 04:51 PM - Ward Vandewege

- Story points set to 0.5

Files

portabledatahash.png	56.3 KB	02/26/2015	Bryan Cosca
portablehash-2.png	30.6 KB	02/26/2015	Bryan Cosca