Arvados - Story #14484

[API Server] Return collection size and number of files in collection record

11/14/2018 04:42 PM - Tom Morris

Status:	Resolved	**Start date:**	03/28/2019
Priority:	Normal	**Due date:**	
Assigned To:	Eric Biagiotti	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2019-04-10 Sprint		

Description

Add database columns (file_count, file_size_total?)

Update in a migration (avoid updating too many rows in a single transaction -- see #13752, af1125bd1)

Use sum of listed file sizes (as opposed to sum of sizes of distinct blocks, etc.)

Subtasks:

Task # 14897: Review 14484-collection-record-update	**Closed**

Related issues:

Related to Arvados - Story #15093: Work with Ops to decide best DB migration ...	**Resolved**

Associated revisions

Revision 08d0b1ab - 04/03/2019 08:42 PM - Eric Biagiotti

Merge remote-tracking branch 'origin/master' into 14484-collection-record-update

refs #14484

Arvados-DCO-1.1-Signed-off-by: Eric Biagiotti <ebiagiotti@veritasgenetics.com>

Revision a048d695 - 04/04/2019 02:54 PM - Eric Biagiotti

Merge branch '14484-collection-record-update'

refs #14484

Arvados-DCO-1.1-Signed-off-by: Eric Biagiotti <ebiagiotti@veritasgenetics.com>

Revision 0d8adf9b - 05/31/2019 06:51 PM - Ward Vandewege

Move the population of the new columns on the collections table to a standalone
script that should be run separate from the migration. Add a note to the
upgrade documentation along those lines. Make the script not blow up on
collections with invalid manifests, but rather just skip them.

refs #15093
refs #14484

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <wvandewege@veritasgenetics.com>

Revision 6f17bfca - 06/03/2019 08:26 PM - Ward Vandewege

Move the population of the new columns on the collections table to a standalone
script that should be run separate from the migration. Add a note to the
upgrade documentation along those lines. Make the script not blow up on
collections with invalid manifests, but rather just skip them.

refs #15093
refs #14484

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <wvandewege@veritasgenetics.com>

History

#1 - 11/14/2018 04:42 PM - Tom Morris

- Status changed from New to In Progress

#2 - 11/21/2018 03:00 PM - Tom Clegg

- Description updated

#3 - 02/20/2019 08:20 PM - Peter Amstutz

- Status changed from In Progress to New

- Target version changed from To Be Groomed to Arvados Future Sprints

- Story points set to 1.0

#4 - 02/27/2019 04:41 PM - Tom Morris

- Target version changed from Arvados Future Sprints to 2019-03-13 Sprint

#5 - 02/27/2019 04:41 PM - Eric Biagiotti

- Assigned To set to Eric Biagiotti

#6 - 03/01/2019 07:33 PM - Tom Morris

- Release set to 15

#7 - 03/13/2019 01:05 PM - Eric Biagiotti

- Target version changed from 2019-03-13 Sprint to 2019-03-27 Sprint

#8 - 03/20/2019 08:59 PM - Eric Biagiotti

- Status changed from New to In Progress

#9 - 03/22/2019 01:42 PM - Eric Biagiotti

Approach:

- Create a migration that will create the columns file_count, file_size_total and populate them with the number of files in the collection (integer DEFAULT 0 NOT NULL) and the sum of the files listed sizes (decimal DEFAULT 0.0 NOT NULL) respectively.
- Update models/collection.rb to include the new columns
- When files are added/removed, update the collections file_count and file_size_total columns.
- Add corresponding INSERT to schema_migrations table in structure.sql
- Create/Modify tests
- Update documentation

Questions:

- During the migration, I can get the files and their sizes from the manifest text. Do I have to parse this manually, or is there an object I should be instantiating?
- When files are added to a collection, where in the code does the manifest get altered? I'm assuming this is the same place that I want to update the [file_count, file_size_total] columns.

#10 - 03/22/2019 02:02 PM - Lucas Di Pentima

Eric Biagiotti wrote:

> - Create a migration that will create the columns file_count, file_size_total and populate them with the number of files in the collection (integer DEFAULT 0 NOT NULL) and the sum of the files listed sizes (decimal DEFAULT 0.0 NOT NULL) respectively.

File size should be in bytes so I suppose we should be using BIGINT instead of DECIMAL?

> - Add corresponding INSERT to schema_migrations table in structure.sql

Check out docs about migrations on rails. The structure.sql file is auto-generated from the migrations. Basically you can write a schema-changing migration first (adding the 2 cols with their default values) and another that does a general update. Then, you run rake db:migrate to apply them and generate the updated structure.sql file.

- Update models/collection.rb to include the new columns
- When files are added/removed, update the collections file_count and file_size_total columns.
- During the migration, I can get the files and their sizes from the manifest text. Do I have to parse this manually, or is there an object I should be instantiating?
- When files are added to a collection, where in the code does the manifest get altered? I'm assuming this is the same place that I want to update the [file_count, file_size_total] columns.

I think the correct strategy would be to write a callback on collection's create & update operations checking if manifest_text changed. When this happens, you scan the new manifest to get the values. This scan code should be added to the Ruby SDK, if it's not already there. This code can be also used from the data migration.

The API Server doesn't support "add/modify file" operations. Those ops are done client-side and then the new resulting manifest is updated on the API server, so that's why you should pay attention when the manifest text changes.

#11 - 03/22/2019 02:29 PM - Peter Amstutz

See https://guides.rubyonrails.org/v4.2/active_record_migrations.html

The magic command is (I have to look this one up every time):

$ bin/rails generate migration TheNameOfTheMigration

A regular 64 bit integer should be fine for file_size_total, no need to use specialized numeric types like decimal or bigint.

You want an active record hook in the Collection model, see https://guides.rubyonrails.org/v4.2/active_record_callbacks.html

You only need to recompute file sizes/counts when "self.manifest_text.changed?"

Use Arv::Collection to parse the manifest text. Then you should be able to iterate over the files.

The migration SHOULD NOT use the Collection class, it should duplicate the logic for computing file counts/sizes (and/or share code in lib/). This is to prevent migrations (which by definition are run on databases created by older versions of the software) from breaking if the behavior of Collection class changes.

#12 - 03/27/2019 12:59 PM - Eric Biagiotti

- Target version changed from 2019-03-27 Sprint to 2019-04-10 Sprint

#13 - 03/28/2019 04:25 PM - Eric Biagiotti

Latest at c5c82ef67b9dc3cb3619e2bef3a86b9b0f0912e8

- Added database migration for adding file_count and file_size_total to the collections table.
- Added logic and test for grouping pdhs by manifest size to the Container model.
- Added file_count and file_size_total to the Collection model.

Unit Tests: https://ci.curoverse.com/view/Developer/job/developer-run-tests/1158/
Integration/Conformance: https://ci.curoverse.com/view/CWL/job/arvados-cwl-conformance-tests/74/console

To verify the migration is working correctly, I just added the following to the end of the up function and ran rake db:migrate:redo STEP=1 RAILS_ENV=test

```
    collections = ActiveRecord::Base.connection.exec_query(
      'SELECT DISTINCT portable_data_hash, manifest_text, file_count, file_size_total FROM collections ORDER B
Y portable_data_hash'
    )
    collections.rows.each do |c|
      print c, "\n"
    end
    print(collections.rows.count)
```

#14 - 03/28/2019 04:50 PM - Eric Biagiotti

Updated documentation at bd2ac2038b13b6ebe92b44cf722c8cf0fa15255b

Unit tests: https://ci.curoverse.com/view/Developer/job/developer-run-tests/1159/

#15 - 03/29/2019 12:34 PM - Lucas Di Pentima

Some comments & questions:

- Why group_pdhs_by_* funcs are part of the Container model? They seem to be more related to Collections as they work with PDHs & manifests, but even then, I think they don't use any of the model's facilities, why not putting them in some separate lib? (didn't follow the discussion on chat)
- There's a typo in a migration comment: "…all the distince pdhs greater…"
- Can you explain the reasoning behind 6709876170511ade8e24fe60bf77da24bc4a03d4 ? I believe that if a manifest isn't valid it should not reach set_file_count_and_total_size() because before_validation :check_manifest_validity, right?
- Could you add collection's model & controller test for this new feature?

#16 - 03/29/2019 12:38 PM - Lucas Di Pentima

Also, please make sure that the new fields are read-only.

#17 - 03/29/2019 06:52 PM - Eric Biagiotti

Lucas Di Pentima wrote:

- There's a typo in a migration comment: "…all the distince pdhs greater…"
- Why group_pdhs_by_* funcs are part of the Container model? They seem to be more related to Collections as they work with PDHs & manifests, but even then, I think they don't use any of the model's facilities, why not putting them in some separate lib? (didn't follow the discussion on chat)

I agree on this. The conclusion of the discussion was that if it is container logic, make a static function on the Container model instead of a lib, even though we try not to use models in migrations. I don't think this is really container logic though.

I moved it and fixed the typo in 59a1fc872723c0bafa9764b95756723f54419631.

Still working on your other comments.

#18 - 04/01/2019 03:37 PM - Eric Biagiotti

- Can you explain the reasoning behind 6709876170511ade8e24fe60bf77da24bc4a03d4 ? I believe that if a manifest isn't valid it should not reach set_file_count_and_total_size() because before_validation :check_manifest_validity, right?

This was a bad attempt at me trying to get tests working. Some tests skip validation, so set_file_count_and_total_size was getting called with invalid manifests. I think the more appropriate fix is to call set_file_count_and_total_size as a part of the after_validation phase. This will ensure that we have a valid manifest when trying to calculate file count and total size.

Fixed in fc636d5e169d944981ce2951e05d59fad04563a3

- Could you add collection's model & controller test for this new feature?

Done

> Also, please make sure that the new fields are read-only.

Done. Any attempt to change the file attributes are overridden with the calculated values from the manifest.

Latest at 79316b0ebbf5bfe934267579478b89f770c0e5ba
Unit tests: https://ci.curoverse.com/view/Developer/job/developer-run-tests/1168/

#19 - 04/03/2019 12:40 PM - Lucas Di Pentima

Some comments:

- On test/unit/collection_test.rb, what I think you're trying to do are collection updates, right? lines 89 & 96 are creating new collections with the passed manifest text, but the comments say "Changing…" If you really meant to make those creations, it would be nice to also check that an update also updates the stats.
- When referencing objects on the fixture I think it's nice not to hardcode their UUIDs on the test, but get them via the helper functions by their fixture name, for example: collections(:fixture_name).uuid
- On models/collection.rb, line 202: I think it's re-computing file sizes & count if a client attempts to change those values. Instead, I think it's cheaper to reassign them to their previous values if they changed but the manifest text didn't. You can access the old value by using for example: self.file_count_was

#20 - 04/03/2019 08:22 PM - Eric Biagiotti

Lucas Di Pentima wrote:

> Some comments:
>
> - On test/unit/collection_test.rb, what I think you're trying to do are collection updates, right? lines 89 & 96 are creating new collections with the passed manifest text, but the comments say "Changing…" If you really meant to make those creations, it would be nice to also check that an update also updates the stats.
> - When referencing objects on the fixture I think it's nice not to hardcode their UUIDs on the test, but get them via the helper functions by their fixture name, for example: collections(:fixture_name).uuid
> - On models/collection.rb, line 202: I think it's re-computing file sizes & count if a client attempts to change those values. Instead, I think it's cheaper to reassign them to their previous values if they changed but the manifest text didn't. You can access the old value by using for example: self.file_count_was

All of the above addressed in 425836b285a32c31ef643f8c5d4b48b8b42b7ac4. I also added a collection controller test for updating a collection with manifest and file stats at the same time in a5cd06261d3ef5005c3bd921c610abfa21dc672f

Unit tests: https://ci.curoverse.com/view/Developer/job/developer-run-tests/1173/

#21 - 04/03/2019 08:40 PM - Lucas Di Pentima

This LGTM, please merge. Thanks!

#22 - 04/04/2019 06:44 PM - Eric Biagiotti

- Status changed from In Progress to Resolved

#23 - 04/10/2019 04:06 PM - Tom Morris

- Related to Story #15093: Work with Ops to decide best DB migration strategy for collection file count & size added

#24 - 09/28/2021 07:04 AM - Roman Jay Almaza

Also, please make sure that the new fields are read-only.[https://bigrigtruckingcompanyaustin.com/[.]]