Arvados - Feature #5622

[Workbench] Fix timeout when generating long "list of collections with this pdh" on content hash page

04/01/2015 05:04 PM - Ward Vandewege

Status:	Resolved	**Start date:**	04/27/2015
Priority:	Normal	**Due date:**	
Assigned To:	Radhika Chippada	**% Done:**	100%
Category:	Workbench	**Estimated time:**	0.00 hour
Target version:	2015-04-29 sprint		

Description

The pathological case is the empty block, for example:

https://workbench.qr1hi.arvadosapi.com/collections/d41d8cd98f00b204e9800998ecf8427e+0

~~We probably need paging here.~~ Moved to #5743

Short term fix: Just limit to some number (20?) and, if there are more results than that, show a message to that effect ("1234 more results not shown"?).

Subtasks:

Task # 5825: Review branch: 5622-fix-timeout-pdh-page	**Resolved**

Related issues:

Related to Arvados - Bug #5743: [Workbench] Replace type-specific #index page...	**New**

Associated revisions

Revision ec0c9f0d - 04/28/2015 06:43 PM - Radhika Chippada

closes #5622
Merge branch '5622-fix-timeout-pdh-page'

History

#1 - 04/01/2015 05:06 PM - Ward Vandewege

- Description updated

#2 - 04/01/2015 05:06 PM - Ward Vandewege

- Target version changed from 2015-04-29 sprint to Bug Triage

#3 - 04/17/2015 06:24 PM - Tom Clegg

- Tracker changed from Bug to Feature

- Subject changed from [Workbench] viewing the content hash page for hashes that are in many projects is too slow to [Workbench] Use infinite scroll for "list of collections with this pdh" on content hash page

- Story points set to 1.0

#4 - 04/17/2015 06:29 PM - Tom Clegg

- Target version changed from Bug Triage to 2015-05-20 sprint

#5 - 04/27/2015 01:33 PM - Radhika Chippada

- Category set to Workbench

- Status changed from New to In Progress

- Assigned To set to Radhika Chippada

- Target version changed from 2015-05-20 sprint to 2015-04-29 sprint

#6 - 04/28/2015 01:44 PM - Brett Smith

Reviewing 5b95344

I'm becoming increasingly concerned that CollectionsController#show is handling too many separate cases. It especially worries me because some of those cases are more like #list than #show. This causes a few problems. First, it means that #show has to duplicate a lot of logic that #list already knows how to do. Second, it means that the PDH result listing doesn't benefit from some of the rich features we have in our listing infrastructure, like sorting. Finally, it means that we have to make special cases in #show views for when we're really doing a portable data hash list—refer to all the places where this branch adds a check on CollectionsHelper::match. I'm concerned that this is going to become unmaintainable in the long term: people may add features to other #show templates without considering the case of PDH listings.

Because the PDH handling is so special, I wonder if this is a good time to break it out the handling into its own logic. Rails routes accept a :constraints option which let you match only when parameters in the URL meet specific criteria. What if we defined a route under collections like get ':id', constraints: {id: /^[0-9a-f]{32}\b/}, action: :show_pdh? Then the logic of the #show_pdh action could just be a conditional redirect: if there's only one collection with that PDH, redirect to #show for that collection by its UUID; otherwise, redirect to #list with the filters parameter filled in with a matching portable_data_hash.

I think this approach would require less code, provide richer collection listings, and be more maintainable over time. What do you think?

Thanks.

#7 - 04/28/2015 02:02 PM - Radhika Chippada

Brett:
I am not comfortable taking up the task of separating this into a new route as part of this bug fix (was initially a bug).

The special handling of pdh display in show was already implemented and other than some minor updates to add infinite scrolling to the page, this is an existing flow.

If we want to add a new route and corresponding logic for pdh page, I think we should handle that as a new feature request.

Are you interested in going ahead and reviewing and fixing this issue with the current flow (which I think is quite a serious performance issue given that it is loading all collections :some 3000+ in this example: + loading each link in a separate api request.

BTW, please pull again. I added comments to infinite_scroll.js so that we can update the "hacking workbench" wiki page as well.

Thanks.

#8 - 04/28/2015 02:54 PM - Tom Clegg

Seems to me the "list of collections with this PDH" tab should be doing an infinite-scroll+filtering enabled collections#index request. If I'm skimming correctly, it looks like the branch in front of us does a collections#show with ?partial=contents_rows, which seems to contain a slightly modified copy of render_index().

AFAICT the idea of overloading the #show action to list collections came up only because the request can sometimes degenerate to a #show, and *used to* even look like a #show. But once you accept that we're really just listing collections at a show-like URI -- /collections/PDH is what "show a collection" URI used to look like a year ago, but now it's shorthand for /collections?filters=[[pdh,=,PDH]] -- it seems a bit confusing to start with a #show controller and bend it until it looks like #index (rather than starting with #index). I think the "http redirect" idea is probably the right way to think about this: in general, /collections/{pdh} is an index request, so treat it like an index. If we detect that we're about to show an index with exactly one entry, redirect to that entry.

I'm not convinced the bugfix merits the complexity (aka maintenance burden, aka future bugs) introduced here...

Couldn't we do all this just by making a separate route (with constraints) -- or even the "if ... return blah" hack at the beginning of #show like we usually do -- with an action that does

1. If zero hits, 404
2. If one hit, redirect to /collections/uuid
3. If >1 hit, redirect to /collections/?filters=[["portable_data_hash","=","foo"]]

This way we'd just reuse all of our existing patterns. (I won't be surprised if something in here won't "just work" ... but what is it? Maybe we can fix it?)

("Sometimes shows a list, sometimes shows an item, depending on stuff" seems rather anti-pattern as an API, and much more likely to create interesting bugs like "#show tab auto-refreshes and surprisingly gets some #index content now because someone copied a collection somewhere": an explicit redirect seems better.)

#9 - 04/28/2015 05:18 PM - Radhika Chippada

From IRC:

tomclegg
12:59 it seems like the "quick fix" here is to stick a limit=20 and an "N more not shown" where we list collections-that-have-this-pdh.

tomclegg
1:04 it seems to me the quick fix is really quick and lets us fix a "this page times out" bug while we work on refactoring stuff in order to make better solutions feasible.
1:07 I'm not really keen on the in-between way where we stuff more code into the area that needs refactoring...

#10 - 04/28/2015 05:33 PM - Tom Clegg

- Description updated

#11 - 04/28/2015 06:21 PM - Radhika Chippada

New branch with the quick fix: 5622-fix-timeout-pdh-page

#12 - 04/28/2015 06:24 PM - Tom Clegg

- Subject changed from [Workbench] Use infinite scroll for "list of collections with this pdh" on content hash page to [Workbench] Fix timeout when generating long "list of collections with this pdh" on content hash page

#13 - 04/28/2015 06:48 PM - Brett Smith

Radhika Chippada wrote:

> New branch with the quick fix: 5622-fix-timeout-pdh-page

19db486 is good to merge. Thanks.

#14 - 04/28/2015 07:05 PM - Radhika Chippada

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

Applied in changeset arvados|commit:ec0c9f0d12560296d4b678c8ad33e1b82221e626.