

## Arvados - Bug #6064

[Workbench] [Performance] Collection#show for /collections/qr1hi-4zz18-hqfxewp0r3dsaov takes forever as it loads all the items for which this collection is the output.

05/18/2015 04:04 PM - Radhika Chippada

|  |                   |                        |                 |
|--|-------------------|------------------------|-----------------|
| <b>Status:</b>   | Resolved          | <b>Start date:</b>     | 05/19/2015      |
| <b>Priority:</b>   | Normal            | <b>Due date:</b>       |                 |
| <b>Assigned To:</b>  | Radhika Chippada  | <b>% Done:</b>         | 100%            |
| <b>Category:</b>   | Workbench         | <b>Estimated time:</b> | 0.00 hour       |
| <b>Target version:</b>   | 2015-05-20 sprint |                        |                 |
| <b>Description</b>   |                   |                        |                 |
| Display of /collections/qr1hi-4zz18-hqfxewp0r3dsaov takes forever.   |                   |                        |                 |
| <ul style="list-style-type: none"><li>The issue seems to be linking all the items under "This collection was the output of". We need to use either infinite scrolling for this or show only a few entries.</li><li>Looking at the wb console log, I noticed the below set of API requests repeated again and again. We must use object caching / preloading for these.</li></ul>   |                   |                        |                 |
| <pre>API client: 6.4034e-05 Prepare request https://qr1hi.arvadosapi.com/arvados/v1/pipeline_instances [{"components", "like", "%qr1hi-4zz18-hqfxewp0r3dsaov%"}] API client: 1.751304491 API transaction API client: 0.006474317 Parse response API client: 0.000310151 Prepare request https://qr1hi.arvadosapi.com/arvados/v1/pipeline_templates {"uuid": "qr1hi-p5p6p-ftcb0o61u4yd2zr"} API client: 0.062476442 API transaction API client: 0.000362192 Parse response API client: 0.000331434 Prepare request https://qr1hi.arvadosapi.com/arvados/v1/pipeline_templates {"uuid": "qr1hi-p5p6p-ftcb0o61u4yd2zr"} API client: 0.059839291 API transaction API client: 0.000241081 Parse response API client: 0.000245865 Prepare request https://qr1hi.arvadosapi.com/arvados/v1/pipeline_templates {"uuid": "qr1hi-p5p6p-ftcb0o61u4yd2zr"} API client: 0.070319694 API transaction API client: 0.000166175 Parse response API client: 0.000262908 Prepare request https://qr1hi.arvadosapi.com/arvados/v1/pipeline_templates {"uuid": "qr1hi-p5p6p-ftcb0o61u4yd2zr"} API client: 0.066435541 API transaction</pre> |                   |                        |                 |
| <b>Subtasks:</b>   |                   |                        |                 |
| Task # 6069: Review branch: 6064-collection-output-of-perf-issue   |                   |                        | <b>Resolved</b> |

### Associated revisions

Revision e5c66d13 - 05/20/2015 01:40 PM - Radhika Chippada

closes #6064

Merge branch '6064-collection-output-of-perf-issue'

### History

#1 - 05/18/2015 04:08 PM - Radhika Chippada

- Description updated

- Category set to Workbench

#2 - 05/18/2015 04:09 PM - Radhika Chippada

- Description updated

#3 - 05/18/2015 06:57 PM - Radhika Chippada

- Status changed from New to In Progress

- Assigned To set to Radhika Chippada

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

In IRC:

tomclegg

1:25 radhika: workbench arvados\_base.rb L102 already uses the rails cache to optimize for cases just like that. Maybe we should look into why it's not being used?

#### #4 - 05/19/2015 09:08 PM - Peter Amstutz

Good catch on the logic for caching. It looks like caching never been used. I wonder what else caching will help with?

However, I'm a little concerned there may be side effects of turning on caching, given that we haven't actually been using it all this time. I don't think there is a default cache timeout, so we might end up with stale objects that don't get updated in a timely manner. Can we do some testing? We will probably need to set `:expires_in` so that objects don't live in the cache for too long. Alternately, we may want to not enable caching in this branch and split the caching fix out into a separate story that can be tested more thoroughly.

The display logic limiting the number of related pipeline instances is fine. It would be nice to have an option to show the entire list, or perform infinite scrolling, but I don't think we need to block on that.

Related, in `_show_source_summary.html.erb` I also noticed:

```
jobs_path(filters: [['log', '=', @object.uuid]].to_json)) do |job| %>
```

Instead of `@object.uuid` it should be `@object.portable_data_hash`

#### #5 - 05/20/2015 01:45 PM - Radhika Chippada

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

- % Done changed from 0 to 100

Applied in changeset arvados|commit:e5c66d135d453a4af4c978590b99a118f65ae12b.