

Arvados Workbench 2 - Bug #15672

List of subprocesses limited in process view

09/30/2019 08:43 AM - Moritz Gilsdorf

Status: Resolved	Start date: 12/16/2019
Priority: Normal	Due date:
Assigned To: Lucas Di Pentima	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version: 2020-01-29 Sprint	
Description	
In order to make large numbers of subprocesses readily available and easily navigable to users:	
<ul style="list-style-type: none">• Switch to using the DataExplorer WB2 component.• Make filtering on state mutually exclusive and use the built in filtering of the DataExplorer. This will require changing the column filter pop up to be radio buttons.• Remove the toggles from the subprocess panel, but keep the data about the number of subprocesses in each state.	
Subtasks:	
Task # 15745: Review 15672-filters-translation-fix	Resolved
Task # 15945: Review 15672-subprocess-list-v2	Resolved
Related issues:	
Related to Arvados - Story #13327: Use new API container_request?include=cont...	New
Related to Arvados - Story #12917: Users should be able to see if a container...	New
Related to Arvados - Story #15350: [Documentation] Add Python SDK recipe on h...	Resolved 06/11/2019
Related to Arvados - Feature #15846: [Workbench2] Data explorer page navigation	New
Related to Arvados - Feature #15844: [Workbench2] Indicate when a DataExplore...	New
Related to Arvados Workbench 2 - Feature #15012: Add /all_processes page	Resolved 01/21/2020
Related to Arvados - Bug #15951: Process filtering UI polish	Resolved 03/04/2020
Related to Arvados - Bug #15902: Very slow query for large workflows	Resolved 01/28/2020
Blocked by Arvados - Feature #15854: Add container request filtering based on...	Resolved 11/20/2019

Associated revisions

Revision 48ae17a5 - 12/17/2019 08:37 PM - Lucas Di Pentima

Merge branch '15672-filters-translation-fix'
Refs #15672

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <lucas@di-pentima.com.ar>

Revision be8de5ac - 01/21/2020 03:24 PM - Lucas Di Pentima

Merge branch '15672-subprocess-list-v2'

Closes #15672

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <lucas@di-pentima.com.ar>

History

#1 - 09/30/2019 08:48 AM - Moritz Gilsdorf

- File Screenshot 2019-09-30 at 10.44.03.png added

There seems to be a limit in how many sub-processes are listed in the process view. In the workflow run shown in the attached screenshot are in total 11169 steps but the api seems to only fetch the first 1000.

!Screenshot 2019-09-30 at 10.44.03.png!

#2 - 10/07/2019 02:26 PM - Tom Morris

- Priority changed from Normal to High

#3 - 10/16/2019 01:59 PM - Tom Morris

- Target version changed from Workbench2 Q3, Q4 to 2019-11-06 Sprint

#4 - 10/23/2019 03:02 PM - Eric Biagiotti

- Assigned To set to Eric Biagiotti

#5 - 10/30/2019 02:56 PM - Peter Amstutz

- Related to Story #13327: Use new API container_request?include=container added

#6 - 10/30/2019 07:12 PM - Tom Morris

- Story points set to 2.0

#7 - 10/30/2019 07:16 PM - Eric Biagiotti

- For listings, use posts with `_method=GET` so we won't get a 414 and load them into state.
- Test performance with 11k+ entries.
- If necessary add virtualization to the grid.

#8 - 10/30/2019 08:17 PM - Eric Biagiotti

- Status changed from New to In Progress

#9 - 11/06/2019 04:12 PM - Eric Biagiotti

- Target version changed from 2019-11-06 Sprint to 2019-11-20 Sprint

#10 - 11/14/2019 09:44 PM - Eric Biagiotti

I am currently working on adding a data explorer to the process/subprocess view (the notion of a process is meant to simplify the container request and container relationship in WB2). The data explorer works by using the list API to only get the data that it needs (# of rows being displayed). Normally, it is trivial to filter because you just update the list query.

But, filtering on process status is presenting a problem because it is based on the container *and* the container request state attribute (e.g. a process is cancelled if the container request priority is 0, or if the container state is "Cancelled").

Example: I want to display 10 rows or subprocesses excluding ones that are cancelled.

- List and filter container_requests where priority != 0, which returns 10 results.
- Use the containerUuid of the returned container_request objects to get the container (if it exists) and filter where state != "Cancelled". Lets say this returns 9 results because one container was cancelled.
- In this state, we don't have enough data to fill all the rows.

The only solution I can think of is to first get all the subprocesses (container requests where requesting_container_uuid is the parent container uuid and the corresponding containers if they exists). Then, I can filter them in the code. But depending on the number of subprocesses and the API request limit, this could add many more network requests and kind of defeats the purpose of the data explorer.

Or maybe I am missing something? Let me know if any of this makes sense. Thanks!

#11 - 11/15/2019 09:06 PM - Eric Biagiotti

A few potential approaches to consider:

Assumptions: All crs have containers. Status filtering operates only on the state/exit_code of the container. Data explorer implementations will have search bars using the trigram index.

Filtering with data explorer

- **Approach:** Get the containerUuid of all the sub-crs that are COMMITTED or FINAL (has a container). This requires adding client functionality to bypass API server response limits. Use the list method to get corresponding containers and filter on the containerUuids and "state"/"exit_code" and limit on number of rows to display.
- **Pros:** Keeps current filtering functionality and adds a data explorer, presenting more useful info to the user.
- **Cons:** Large network requests, scalability problems.

WB1 style with no filtering and infinite scroll

- **Approach:** Remove filtering on status and put the data explorer in infinite scroll mode. This adds a button at the bottom of the page that says "Load More" and is more like how WB1 works. This could still result in rendering performance issues unless we block this behind the react-redux update and try using list virtualization.
- **Pros:** Network friendly and more scalable.
- **Cons:** Removes functionality from the process page. Could still result in rendering issues with a full list displayed.

New API feature

- **Approach:** Block this behind adding an API feature to be able to filter container requests by their container's state and exit_code.
- **Pros:** Network friendly and more scalable. Data explorer will work well and no filtering needs to be done on the client side. Users keep existing filters.
- **Cons:** API work has to be done first.

#12 - 11/15/2019 09:33 PM - Peter Amstutz

Eric Biagiotti wrote:

A few potential approaches to consider:

Assumptions: All crs have containers. Status filtering operates only on the state/exit_code of the container. Data explorer implementations will have search bars using the trigram index.

Filtering with data explorer

- **Approach:** Get the containerUuid of all the sub-crs that are COMMITTED or FINAL (has a container). This requires adding client functionality to bypass API server response limits. Use the list method to get corresponding containers and filter on the containerUuids and "state"/"exit_code" and limit on number of rows to display.
- **Pros:** Keeps current filtering functionality and adds a data explorer, presenting more useful info to the user.
- **Cons:** Large network requests, scalability problems.

WB1 style with no filtering and infinite scroll

- **Approach:** Remove filtering on status and put the data explorer in infinite scroll mode. This adds a button at the bottom of the page that says "Load More" and is more like how WB1 works. This could still result in rendering performance issues unless we block this behind the react-redux update and try using list virtualization.
- **Pros:** Network friendly and more scalable.
- **Cons:** Removes functionality from the process page. Could still result in rendering issues with a full list displayed.

New API feature

- **Approach:** Block this behind adding an API feature to be able to filter container requests by their container's state and exit_code.
- **Pros:** Network friendly and more scalable. Data explorer will work well and no filtering needs to be done on the client side. Users keep existing filters.
- **Cons:** API work has to be done first.

Maybe we should have a quick meeting on this next week but it feels to me like doing it all client side is actually a nontrivial amount of work for an inferior result, whereas if we add an API side query that does exactly what we want (which I believe would be relatively easy), we will also be able to plug it into the existing data table with minimal effort.

#13 - 11/15/2019 09:57 PM - Tom Morris

- Related to Story #12917: Users should be able to see if a container failed from the container_request method added

#14 - 11/15/2019 09:57 PM - Tom Morris

- Related to Story #15350: [Documentation] Add Python SDK recipe on how to get a container request's real state added

#15 - 11/19/2019 09:50 PM - Eric Biagiotti

- Description updated

#16 - 11/20/2019 03:38 PM - Eric Biagiotti

- Target version changed from 2019-11-20 Sprint to 2019-12-04 Sprint

#17 - 11/20/2019 04:48 PM - Eric Biagiotti

- Description updated

#18 - 11/20/2019 04:53 PM - Eric Biagiotti

- Blocked by Feature #15854: Add container request filtering based on container.exit_code and container.state added

#19 - 11/20/2019 05:10 PM - Eric Biagiotti

- Related to Feature #15846: [Workbench2] Data explorer page navigation added

#20 - 11/20/2019 05:10 PM - Eric Biagiotti

- Related to Feature #15844: [Workbench2] Indicate when a DataExplorer filter is active added

#21 - 12/04/2019 03:50 PM - Eric Biagiotti

- Target version changed from 2019-12-04 Sprint to 2020-01-02 Sprint

#22 - 12/09/2019 09:41 PM - Eric Biagiotti

- Docs updated @ commit:d953b5510676f1cb5cfab7a092d2de2cd3c72135 in the Arvados repo.

Latest work:

- Added the ability to create radio button groups for data explorer column filtering.
- Process panel now stores the parent container_request uuid, which is used as a prop to the subprocess panel to populate the dataexplorer.
- Added Run Time and Created At columns. Can't currently sort by runtime.

TODO:

- Add started_at and finished_at time to columns. Users can use the column filtering on the top right of the data explorer to choose what columns to display. Pick a reasonable set of defaults.
- The action triple dot for subprocesses in the data explorer is not set up. Look at project-panel.tsx for an example.
- Search of the subprocess view from the data explorer doesn't work. Use trigram index API call for this.
- Default message on the data explorer for when there are no subprocesses in the view is not being displayed.
- Clicking on a subprocess in the data explorer should update the details panel.
- Navigation when double clicking on a subprocess isn't working.
- Reset the data explorer radio box when navigating to the page?

#23 - 12/11/2019 05:08 PM - Lucas Di Pentima

- Assigned To changed from Eric Biagiotti to Lucas Di Pentima

#24 - 12/12/2019 03:43 PM - Peter Amstutz

- Related to Feature #15012: Add /all_processes page added

#25 - 12/16/2019 10:15 PM - Lucas Di Pentima

Filters were being constructed by doing snake case translation on the attribute name.

Branch 15672-filters-translation-fix at commit 680b1ffd (wb2 repo) removed this translation and fixes all parts of the code using the FilterBuilder class.

This is needed on this branch because the filter `[[container.state', '=', 'foo]]` gets translated to `[[container_state', '=', 'foo]]`.

Also found a similar case on ordering, but that is being type checked against the resources' key names so I think it's not so trivial to fix.

#26 - 12/17/2019 06:15 PM - Peter Amstutz

Lucas Di Pentima wrote:

Filters were being constructed by doing snake case translation on the attribute name.

Branch 15672-filters-translation-fix at commit 680b1ffd (wb2 repo) removed this translation and fixes all parts of the code using the FilterBuilder class.

This is needed on this branch because the filter `[[container.state', '=', 'foo]]` gets translated to `[[container_state', '=', 'foo]]`.

Also found a similar case on ordering, but that is being type checked against the resources' key names so I think it's not so trivial to fix.

This LGTM.

#27 - 12/20/2019 11:38 PM - Lucas Di Pentima

Updates at commit 134e0d36 - branch 15672-subprocess-list-v2 (wb2 repo)

Continued Eric's branch on a separate one, having rebased it to use the previous fixes on filter translation:

- Made status filters added by Eric to be used on the request
- Pagination options now offer 50, 100, 200 and 500 page sizes
- Removed subprocess count and filter sliders panel on the top right corner
- Double click and name-click navigate to the selected subprocess
- Single click updates the detail panel
- When no subprocesses available to list (because of filtering, search, etc) show a message.
- Modified the list api calls to use POST only when the params on the query string are big enough (>1500 chars)
- Added subprocess search by name, not sure if we should be able to search by other properties

#28 - 12/30/2019 04:34 PM - Peter Amstutz

Functional comments, haven't looked at the code yet:

Locked and running containers have a runtime of "NaNh NaNm NaNs" (running containers should compute it as now - started_at, Locked should be either blank or zero)

In the "Status filters" popup, clicking on the button works to change the filter. However moving the mouse cursor over the text of an item and clicking it changes the item background but doesn't change the filter. Also it seems unnecessary to select the filter and then separately choose Ok/Cancel, a single click should select the item and close the dialog.

After closing the "Status filters" popup, the "Filters" tooltip remains on the screen, partly obscuring the first row.

The list live-updates the state of individual subprocess items but does not refresh the list. So if you filter on "Running" the items in the list can live-update to "Completed" but are not removed from the list, and new "Running" subprocesses are not added. There's also no button to explicitly refresh the list (this something that is missing from the UI design in general, clicking the item in the breadcrumb bar usually works but is not at all obvious).

#29 - 12/30/2019 06:30 PM - Peter Amstutz

Code @ 16860c6f2fedc5c150cab796e93aa21067f28e97

subprocess-panel-actions.ts has a bunch of commented-out code

I see there's a ContainerRunTime component so it shouldn't be too hard to fix the time display (although since 'now' constantly changes, I'm not sure about the preferred way to handle that sort of thing in React, we probably only want update granularity to the nearest minute).

#30 - 12/30/2019 06:40 PM - Peter Amstutz

Switching from GET to POST when the URI gets too long -- could we unit test that?

#31 - 12/30/2019 06:42 PM - Peter Amstutz

- Related to Bug #15951: Process filtering UI polish added

#32 - 01/02/2020 03:46 PM - Peter Amstutz

- Target version changed from 2020-01-02 Sprint to 2020-01-15 Sprint

#33 - 01/07/2020 03:10 PM - Peter Amstutz

- Assigned To changed from Lucas Di Pentima to Peter Amstutz

#34 - 01/15/2020 03:47 PM - Peter Amstutz

- Target version changed from 2020-01-15 Sprint to 2020-01-29 Sprint

#35 - 01/15/2020 04:35 PM - Peter Amstutz

- Assigned To changed from Peter Amstutz to Lucas Di Pentima

#36 - 01/16/2020 08:20 PM - Tom Clegg

- Related to Bug #15902: Very slow query for large workflows added

#37 - 01/18/2020 02:16 AM - Lucas Di Pentima

Updates at commit 6183e02

- Reviewed Peter's additions and fixed a broken test
- Added a unit test for the list API call with POST method when using big query params as requested on note-30

#38 - 01/18/2020 11:48 AM - Lucas Di Pentima

Simplified test code @ commit 22a3f545

#39 - 01/21/2020 02:54 PM - Peter Amstutz

Lucas Di Pentima wrote:

Simplified test code @ commit 22a3f545

My only comment would be to recommend having two test cases, one just under the limit (uses GET) and one just over the limit (uses POST), currently only tests the second case one so every query could use POST and this wouldn't detect it.

Rest LGTM, please merge.

#40 - 01/21/2020 03:07 PM - Lucas Di Pentima

Added test @ commit ea35db0f

#41 - 01/21/2020 03:29 PM - Anonymous

- % Done changed from 50 to 100

- Status changed from In Progress to Resolved

Applied in changeset [arvados-workbench2|be8de5acc35c47d0b191223e5ece96fcf452ea5d](#).

#42 - 01/14/2021 04:26 PM - Peter Amstutz

- Related to Story #16945: WB2 Workflows / containers feature parity added

#43 - 01/14/2021 04:27 PM - Peter Amstutz

- Related to deleted (Story #16945: WB2 Workflows / containers feature parity)

Files

Screenshot 2019-09-30 at 10.44.03.png	286 KB	09/30/2019	Moritz Gilsdorf
---------------------------------------	--------	------------	-----------------