

## Arvados - Bug #15902

### Very slow query for large workflows

12/03/2019 03:32 PM - Tom Morris

<b>Status:</b>	Resolved	<b>Start date:</b>	01/28/2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Tom Clegg	<b>% Done:</b>	100%
<b>Category:</b>	API	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2020-01-29 Sprint		

#### Description

A customer reports that the query below takes 32 seconds for a ~3,000 container workflow:

```
duration: 32570.970 ms execute &lt;unnamed&gt;: SELECT container_requests."uuid", container_requ
ests."owner_uuid", container_requests."created_at", container_requests."modified_at", container_re
quests."modified_by_client_uuid", container_requests."modified_by_user_uuid", container_requests."
name", container_requests."description", container_requests."properties", container_requests."stat
e", container_requests."requesting_container_uuid", container_requests."container_uuid", container
_requests."container_count_max", container_requests."runtime_constraints", container_requests."con
tainer_image", container_requests."environment", container_requests."cwd", container_requests."com
mand", container_requests."output_path", container_requests."priority", container_requests."expire
s_at", container_requests."filters", container_requests."container_count", container_requests."use
_existing", container_requests."scheduling_parameters", container_requests."output_uuid", containe
r_requests."log_uuid", container_requests."output_name", container_requests."output_ttl"
FROM "container_requests"
```

```
WHERE (container_requests.owner_uuid NOT IN (SELECT target_uuid FROM materialized_permission_view
WHERE trashed = 1) ) AND (container_requests.requesting_container_uuid in ( '
zzzzz-dz642-r9n91uyk20xowyf',...[~3000 UUIDs omitted]
) ORDER BY container_requests.modified_at desc, container_requests.uuid LIMIT $1 OFFSET $2
< 2019-11-26 10:50:27.162 EST >DETAIL: parameters: $1 = '1000', $2 = '0'
```

Here's the EXPLAIN ANALYZE for the original query:

```
Limit (cost=2385.09..16840.05 rows=1000 width=791) (actual time=32385.658..32385.658 rows=0 loops
=1)
-> Index Scan using index_container_requests_on_modified_at_uuid on container_requests (cost=
2385.09..1995593.21 rows=137891 width=791) (actual time=32385.656..32385.656 rows=0 loops=1)
Filter: ((NOT (hashed SubPlan 1)) AND ((requesting_container_uuid)::text = ANY ('{
zzzzz-dz642-r9n91uyk20xowyf,...}':::text[])))
Rows Removed by Filter: 376597
SubPlan 1
-> Index Only Scan using permission_target_trashed on materialized_permission_view (c
ost=0.42..2380.40 rows=1709 width=28) (actual time=0.050..0.351 rows=95 loops=1)
Index Cond: (trashed = 1)
Heap Fetches: 95
Planning time: 29.160 ms
Execution time: 32385.751 ms
```

Deleting the index that the query planner incorrectly chose, `index_container_requests_on_modified_at_uuid`, dropped the query time from 30+ seconds to ~50 msec.

```
Limit (cost=83535.57..83538.07 rows=1000 width=780) (actual time=54.419..54.419 rows=0 loops=1)
-> Sort (cost=83535.57..83877.22 rows=136657 width=780) (actual time=54.417..54.417 rows=0 lo
ops=1)
Sort Key: container_requests.modified_at DESC, container_requests.uuid
Sort Method: quicksort Memory: 25kB
-> Index Scan using index_container_requests_on_requesting_container_uuid on container_r
equests (cost=635.83..76042.82 rows=136657 width=780) (actual time=54.323..54.323 rows=0 loops=1)
```

Before this was done, a full VACUUM ANALYZE had no affect on performance, but dropping the ORDER BY clause from the query also dramatically improved the query times.

One thing that seems strange is that the query is ordering by not only modified\_at, but also UUID which doesn't really seem to add

any value.

We should see if we can adjust the indexes and/or query in a way which the query planner can optimize correctly.

**Subtasks:**

Task # 15966: Review 15902-slow-query

**Resolved**

**Related issues:**

Related to Arvados Workbench 2 - Bug #15672: List of subprocesses limited in ...

**Resolved**

**12/16/2019**

**Associated revisions**

**Revision 40501735 - 01/28/2020 10:06 PM - Lucas Di Pentima**

Merge branch '15902-slow-query'

Closes #15902

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

**History**

**#1 - 12/04/2019 04:33 PM - Tom Morris**

- Assigned To set to Ward Vandewege

**#2 - 01/02/2020 03:49 PM - Peter Amstutz**

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

**#3 - 01/02/2020 03:59 PM - Peter Amstutz**

- Assigned To deleted (Ward Vandewege)

**#4 - 01/02/2020 04:18 PM - Peter Amstutz**

- Assigned To set to Peter Amstutz

**#5 - 01/02/2020 07:04 PM - Peter Amstutz**

- Category set to API

**#6 - 01/08/2020 05:08 PM - Tom Clegg**

- Release set to 22

**#7 - 01/08/2020 05:09 PM - Tom Clegg**

- Assigned To changed from Peter Amstutz to Tom Clegg

**#8 - 01/14/2020 07:50 PM - Tom Clegg**

- Status changed from New to In Progress

Deleting the (modified\_at desc, uuid) index doesn't seem appealing -- it's likely to make different queries slow.

It may be possible to tune postgresql's statistics / query planner so it chooses the appropriate index.

I suspect the easiest short term solution/workaround is for the client to specify order=uuid or order=(requesting\_container\_uuid, uuid), rather than using the default order=(modified\_at desc, uuid).

I'm guessing the client is workbench or workbench2?

**#9 - 01/15/2020 03:50 PM - Peter Amstutz**

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

**#10 - 01/16/2020 08:20 PM - Tom Clegg**

- Related to Bug #15672: List of subprocesses limited in process view added

**#11 - 01/16/2020 09:36 PM - Lucas Di Pentima**

I have checked wb2 code and I haven't found code making a request with a 'requesting\_container\_uuid in [some list]' filter. In fact, current wb2's master branch would fail to show a big workflow because it makes a request with a filter too big to be used on a GET query.

**#12 - 01/28/2020 08:10 PM - Tom Clegg**

15902-slow-query @ [9fc2ff5583fc31c9f75f80dcf7258bed63084568](https://ci.arvados.org/view/Developer/job/developer-run-tests/1710/) -- <https://ci.arvados.org/view/Developer/job/developer-run-tests/1710/>

**#13 - 01/28/2020 08:36 PM - Lucas Di Pentima**

This LGTM, waiting for tests to finish before merging.

**#14 - 01/28/2020 09:28 PM - Lucas Di Pentima**

Re-running wb integration tests just in case

<https://ci.arvados.org/job/developer-run-tests-apps-workbench-integration/1811/>

**#15 - 01/28/2020 10:51 PM - Anonymous**

- *Status changed from In Progress to Resolved*

Applied in changeset [arvados|405017353d0a74ef28288744957fd362df98c0e2](https://ci.arvados.org/view/Developer/job/developer-run-tests-apps-workbench-integration/1811/).