Arvados - Story #12085

Add monitoring/alarm for failed/slow job dispatch & excess idle nodes

08/08/2017 02:02 PM - Tom Morris

Status:	Resolved	**Start date:**	08/08/2017
Priority:	Normal	**Due date:**	
Assigned To:	Lucas Di Pentima	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2018-04-11 Sprint		

Description

We need some additional monitoring and alarms to catch situations like yesterday's crunch-dispatch.rb file descriptor issue.

Some suggestions for alarm conditions:

- more than N (15? 15% of running_nodes?) idle nodes for more than M (10?) minutes
- jobs queued for more than 15 minutes when there is idle capacity in the cluster (running_nodes < 0.95 * max_nodes)

The thresholds, sampling periods, and triggers periods can be adjusted as we gain experience with what's too little or too much. The goal is to ignore brief transients or normal steady state churn, but quickly (< 1 hr) catch abnormal conditions which otherwise take us hours to notice on an ad hoc basis.

Subtasks:

Task # 13224: Review 12085-anm-metrics	**Resolved**

Related issues:

Related to Arvados - Story #11836: [Nodemanager] Improve status.json for moni...	**Rejected**	**05/23/2018**

Associated revisions

Revision 2fbdfebf - 04/05/2018 03:15 PM - Lucas Di Pentima

Merge branch '12085-anm-metrics'
Closes #12085

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

History

#1 - 08/16/2017 07:06 PM - Tom Morris

- Target version changed from 2017-08-16 sprint to 2017-08-30 Sprint

#2 - 08/30/2017 07:14 PM - Tom Morris

- Target version changed from 2017-08-30 Sprint to 2017-09-13 Sprint

#3 - 08/30/2017 07:37 PM - Tom Morris

- Target version changed from 2017-09-13 Sprint to 2017-09-27 Sprint

#4 - 09/14/2017 08:47 PM - Tom Morris

- Target version changed from 2017-09-27 Sprint to 2017-10-11 Sprint

#5 - 09/19/2017 06:28 PM - Tom Clegg

some ideas for metrics (mostly implemented in nodemanager)

- max queue time of any queued job/container (this should be implemented in apiserver, not nodemanager)
- number of alive compute nodes
- number of allocated compute nodes
- configured max compute nodes
- total hourly cost
- configured max hourly cost
- are we currently waiting for a node to turn off before trying again because we hit a quota?
- max idle time of any compute node
- max uptime of any compute node
- number of errors received from cloud provider by this process

- this process uptime
- number of occurrences of unpaired→shutdown transition (node was created but never pinged within configured boot-wait)

#6 - 09/19/2017 06:51 PM - Tom Morris

- Story points set to 1.0

#7 - 09/27/2017 07:20 PM - Tom Morris

- Project changed from OPS to Arvados

- Target version changed from 2017-10-11 Sprint to Arvados Future Sprints

#8 - 03/14/2018 07:42 PM - Tom Morris

- Target version changed from Arvados Future Sprints to 2018-03-28 Sprint

#9 - 03/14/2018 07:45 PM - Lucas Di Pentima

- Assigned To set to Lucas Di Pentima

#10 - 03/15/2018 03:00 PM - Lucas Di Pentima

- Status changed from New to In Progress

#11 - 03/15/2018 03:32 PM - Nico César

from note-5 I'll be ordering by priority for Ops Needs and the reason in parenthesis:

1. configured max compute nodes (this will help on all graphs, basically exposing the configuration)
2. idle time of all compute node (we can get the max of this, but also we need to know which idle nodes are misbehaving)
3. number of alive compute nodes and number of allocated compute nodes (this 2 metrics will give us an idea current state of costs versus actual use)
4. number of occurrences of unpaired→shutdown transition (node was created but never pinged within configured boot-wait)
5. number of errors received from cloud provider by this process
6. number of exceptions generated by all actors in this process (and successfully catched)
7. this process uptime (this is kind of irrelevant since the process will suicide if a major problem happens but also in a normal deploy. If it's easy to do, then is better to have it, otherwise just ignore)

Anything related to costs we're implementing it separately today . it will a "nice to have" but again, not urgent

1. total hourly cost
2. configured max hourly cost

this is not relevant in my opinion or it's unclear to me what we want to achive

1. max uptime of any compute node
2. are we currently waiting for a node to turn off before trying again because we hit a quota?

Queue times are important, but simplifying in a max aggregate could be hiding important things, I would leave this out of this ticket and rethink what we want here.

1. max queue time of any queued job/container (this should be implemented in apiserver, not nodemanager)

#12 - 03/28/2018 03:12 PM - Lucas Di Pentima

- Target version changed from 2018-03-28 Sprint to 2018-04-11 Sprint

#13 - 03/28/2018 08:46 PM - Lucas Di Pentima

Updates at a02012dd9 - branch 12085-anm-metrics
Test run: https://ci.curoverse.com/job/developer-run-tests/671/

New metrics on nodemanager status tracker:

- max_nodes: Expose nodemanager's configuration
- actor_exceptions: Actor non fatal error counter
- cloud_errors: CLOUD_ERRORS exception counter
- boot_failures: Number of times any node goes from unpaired to shutdown
- idle_nodes: Hash with counters for every node that is on idle state, stating how many seconds is in that state. When a node leaves the idle state, it's removed from this hash (asked Nico about this behavior).

Also added tests for all the new stats.

Regarding the number of alive versus allocated nodes, the status tracker already show how many nodes are on every state, so I think that's enough.

#14 - 03/30/2018 04:03 PM - Peter Amstutz

In addition to max_nodes it should also expose the current value of node_quota.

Could we make cloud_errors more specific? Like "create_node_errors", "destroy_node_errors", "list_node_errors" ?

It looks like it is missing a call to idle_out() when a node disappears from the cloud node list?

#15 - 03/30/2018 05:22 PM - Peter Amstutz

Would it make sense to add a counter like "time_spent_idle" which is the sum of node idle times since node manager started? It might be useful to get a sense of how much time is actually wasted.

#16 - 03/30/2018 07:16 PM - Nico César

Peter Amstutz wrote:

> Would it make sense to add a counter like "time_spent_idle" which is the sum of node idle times since node manager started? It might be useful to get a sense of how much time is actually wasted.

I think all agregations can be done in the tool that uses this data. Even if is a trivial thing in the code, I feel there will be turtles down the road when node manager restarts (which suicide is the mother of all fallbacks in a-n-m)

#17 - 04/03/2018 05:00 PM - Lucas Di Pentima

Updates at c18fb83a3
Test run: https://ci.curoverse.com/job/developer-run-tests/675/

- Added node_quota metrics.
- Splitted cloud_errors into list_nodes_errors, create_node_errors and destroy_node_errors.
- Added missing status.tracker.idle_out() call when a idle node is detected to be missing from the cloud node list.
- Added/updated related tests.

#18 - 04/05/2018 02:19 PM - Lucas Di Pentima

Rebased against latest master at 5fd2ed9e93670007226a1772040a966fb9dd4d22
Test run: https://ci.curoverse.com/job/developer-run-tests/676/

#19 - 04/05/2018 02:22 PM - Peter Amstutz

```
                if record.actor:
                    try:
                        # If it's paired and idle, stop its idle time counter
                        # before removing the monitor actor.
                        if record.actor.get_state().get() == 'idle':
                            status.tracker.idle_out(
                                record.actor.arvados_node.get()['hostname'])
                        record.actor.stop()
                    except pykka.ActorDeadError:
                        pass
```

Take out if record.actor.get_state().get() == 'idle': and call it unconditionally.

I believe you can use record.arvados_node["hostname"] directly intead of calling the actor.

Actually I think the whole block should go outside of "if record.actor"

```
if record.arvados_node:
    status.tracker.idle_out(record.arvados_node.get('hostname'))
if record.actor:
    ...
```

#20 - 04/05/2018 02:39 PM - Lucas Di Pentima

Suggestion addressed at 842c85cde
Test run: https://ci.curoverse.com/job/developer-run-tests/677/

#21 - 04/05/2018 02:57 PM - Peter Amstutz

Lucas Di Pentima wrote:

> Suggestion addressed at 842c85cde
> Test run: https://ci.curoverse.com/job/developer-run-tests/677/

LGTM

#22 - 04/05/2018 03:41 PM - Lucas Di Pentima

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

Applied in changeset arvados|2fbdfebf757e5a9b53cf0a21facdf2bd3ea6c757.

#23 - 07/23/2018 07:00 PM - Tom Morris

- Release set to 13