Orchestrator - Bug #37514

mgr CLI commands block one another (indefinitely if the orchestrator CLI gets stuck in wait)

12/04/2018 06:46 AM - Tim Serong

Status:	Can't reproduce	**% Done:**	0%
Priority:	Normal		
Assignee:			
Category:			
Target version:			
Source:		**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:		**Pull request ID:**	
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	
Reviewed:			

Description

There seems to be two problems here:

1) Only one mgr CLI command runs at a time. This isn't obvious unless you find an mgr command that takes a few seconds to run. For example, when testing the deepsea orchestrator, `ceph orchestrator device ls` takes about five seconds to complete. If I invoke that command in one terminal window, then invoke `ceph osd status` in another terminal window, the latter will block until the former completes. That's irritating, but probably not fatal.

2) Orchestrator CLI commands spin waiting for command completions from whatever orchestrator module is active. If you manage to break an orchestrator in such a way as commands never complete (try e.g.: stopping the salt-api while using DeepSea), `ceph orchestrator device ls` will never complete. Even if you hit CTRL-C, mgr is (presumably) still stuck in that loop waiting for completions that are never going to happen, which means it's unable to service any subsequent CLI command. Trying to run, say, `ceph osd status` at this point will also simply just hang. You can't even quickly restart mgr at this point: `systemctl restart ceph-mgr$(hostname)`@ hangs until it hits "State 'stop-sigterm' timed out." (after maybe a minute and a half), then it sends a SIGKILL and mgr is finally restarted.

Related issues:

Related to Orchestrator - Bug #43153: mgr orchestrator hangs	**Can't reproduce**
Related to Orchestrator - Feature #39093: mgr/orchestrator: add `ceph orchest...	**Rejected**

History

#1 - 12/04/2018 07:24 PM - Juan Miguel Olmo Martínez

1) I think this is due the implementation that we are using in this moment. Although the "wait" method in each of the orchestrator modules can work with several operations simultaneously, we are sending (orchestrator_cli._wait) a list with only one completion object in each different operation. I think this makes that operations are executed sequentially.
I have not verified but probably if we "accumulate" in the "completions list" of the orchestrator the different operations sent by the orchestrator_cli, this operations will run apparently "simultaneously".
In any case, there is an open discussion about if we should allow to run operations simultaneously or not, maybe certain operations (e.g. osd creation) must run sequentially, while others (status, device ls) can be executed in "parallel"

2) I have verified, that in the ansible case this is not the behavior. In my case i have my external orchestrator running in a container. i issue a "device ls" command and i "pause" the external orchestrator container. In my casee there is no return for the command , but if i press "ctrl+c" the command is stopped and a message "^CError EINTR: Interrupted!" appears.
If instead of "ctrl-c" i "unpause" the external orchestrator container, then the command continues and the output is correct.
In both cases, (ctrl+c, unpause) if i try another command, it works ok.

Note: I have made the test with the latest "master"... check if you have this line(orchestrator_cli._wait):
done = self._oremote("wait", completions) == []

I have verified that "self._oremote("wait", completions)" is not evaluated as "False" when completions list is empty (i have suspicions about how information is passed between python interpreters in the mgr module). This is why i added the apparently redundant comparation with the empty list.

#2 - 12/05/2018 04:51 AM - Tim Serong

It's not just orchestrator CLI commands -- it's all mgr module commands that are forced to execute sequentially. Try issuing "device ls", then pause your external orchestrator, press CTRL+C, then try to run a non-orchestrator mgr CLI command, e.g.: `ceph osd status`. In my testing, this unrelated command will also block, because mgr is stuck waiting for the orchestrator command to complete.

#3 - 02/25/2019 10:23 AM - Sebastian Wagner

- Category set to orchestrator

#4 - 08/22/2019 12:10 PM - Sebastian Wagner

my 2 cents:

regarding 1)

We have to distinguish between the CLI commands and the orchestrator implementations:

Right, only one CLI can run at a time, until https://github.com/ceph/ceph/pull/26565 is merged. On the other hand, we can already have multiple orchestrator module invocations, by e.g. the dashbaord, because the Orchestrator.wait method doesn't wait! So, you can have multiple completions simultaneous if orchestrators don't actually wait in wait().

regarding 2)

Here, I'd propose something like a ceph orchestrator clear that unstucks completions. See https://trello.com/c/S5HbDXjP/117-ceph-orchestrator-clear

#5 - 10/29/2019 04:44 AM - Tim Serong

It's not just orchestrator related CLI commands. AFAICT any CLI command from any module can block any CLI command from any other module.

#6 - 01/16/2020 03:49 PM - Sage Weil

- Project changed from mgr to Orchestrator

- Category deleted (orchestrator)

#7 - 01/21/2020 11:22 AM - Sebastian Wagner

- Related to Bug #43153: mgr orchestrator hangs added

#8 - 01/21/2020 11:22 AM - Sebastian Wagner

- Related to Feature #39093: mgr/orchestrator: add `ceph orchestrator wait` added

#9 - 03/12/2020 12:37 PM - Sebastian Wagner

- Status changed from New to Can't reproduce

CLI commands should now respond swiftly. (cephadm and rook)