

Ceph - Bug #8797

"ceph status" do not exit with python_2.7.8

07/09/2014 11:30 PM - Dmitry Smirnov

Status:	Resolved	Start date:	07/09/2014
Priority:	High	Due date:	
Assignee:	Dan Mick	% Done:	0%
Category:	ceph cli	Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Source:	Community (user)	Reviewed:	
Tags:		Affected Versions:	
Backport:	firefly,giant	ceph-qa-suite:	
Regression:	No	Pull request ID:	
Severity:	2 - major	Crash signature:	
Description			
As reported in https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=754341 after upgrade to python 2.7.8 "ceph -s`" hangs instead of returning to shell.			
Related issues:			
Related to Ceph - Bug #10567: Python loctx should retain a reference to the c...		Resolved	01/18/2015

Associated revisions

Revision 5ba9b8f2 - 12/10/2014 09:19 PM - Dan Mick

rados.py: remove Rados.__del__(); it just causes problems

Recent versions of Python contain a change to thread shutdown that causes ceph to hang on exit; see <http://bugs.python.org/issue21963>. As it turns out, this is relatively easy to avoid by not spawning threads on exit, as Rados.__del__() will certainly do by calling shutdown(); I suspect, but haven't proven, that the problem is that shutdown() tries to start() a threading.Thread() that never makes it all the way back to signal start().

Also add a PendingReleaseNote and extra doc comments to clarify.

Fixes: #8797

Signed-off-by: Dan Mick <dan.mick@redhat.com>

Revision ed8c9af3 - 12/11/2014 05:29 AM - Dan Mick

rados.py: remove Rados.__del__(); it just causes problems

Recent versions of Python contain a change to thread shutdown that

causes ceph to hang on exit; see <http://bugs.python.org/issue21963>.

As it turns out, this is relatively easy to avoid by not spawning threads on exit, as Rados.__del__() will certainly do by calling shutdown(); I suspect, but haven't proven, that the problem is that shutdown() tries to start() a threading.Thread() that never makes it all the way back to signal start().

Also add a PendingReleaseNote and extra doc comments to clarify.

Fixes: #8797

Signed-off-by: Dan Mick <dan.mick@redhat.com>

(cherry picked from commit 5ba9b8f21f8010c59dd84a0ef2acfec99e4b048f)

Conflicts:

PendingReleaseNotes

Revision e00270b5 - 12/11/2014 05:35 AM - Dan Mick

rados.py: remove Rados.__del__(); it just causes problems

Recent versions of Python contain a change to thread shutdown that causes ceph to hang on exit; see <http://bugs.python.org/issue21963>.

As it turns out, this is relatively easy to avoid by not spawning threads on exit, as Rados.__del__() will certainly do by calling shutdown(); I suspect, but haven't proven, that the problem is that shutdown() tries to start() a threading.Thread() that never makes it all the way back to signal start().

Also add a PendingReleaseNote and extra doc comments to clarify.

Fixes: #8797

Signed-off-by: Dan Mick <dan.mick@redhat.com>

(cherry picked from commit 5ba9b8f21f8010c59dd84a0ef2acfec99e4b048f)

Conflicts:

PendingReleaseNotes

History

#1 - 07/10/2014 02:17 PM - Dmitry Smirnov

- Priority changed from Normal to High

Looks like librados thread is active (not terminated) in "rados.py" and `ceph` is waiting for it indefinitely... I'm illiterate in Python so I can't analyse further...

#2 - 07/10/2014 08:05 PM - Dmitry Smirnov

- Priority changed from High to Normal

This bug prevent Ceph daemons from starting...
Debian "testing" is already affected since python already propagated there...
Please advise.

#3 - 07/10/2014 08:06 PM - Dmitry Smirnov

- Priority changed from Normal to High

#4 - 07/11/2014 09:00 AM - Dmitry Smirnov

Please be advised that this issue appears to be a regression in Python 2.7.8 (see details in the Debian bug report).

#5 - 07/15/2014 02:40 AM - Dmitry Smirnov

<http://bugs.python.org/issue21963>

#6 - 07/15/2014 12:20 PM - Dan Mick

Fascinating info so far, Dmitry, thanks for your work on this. Anxious to see what the Python team thinks of the assertion that it's a Python issue.

#7 - 07/16/2014 05:23 AM - Alfredo Deza

I believe that we should attempt to replicate the problem first as I know the Python ticket will get ignored unless there is a way to reproduce (other than installing ceph with that Python version and call `ceph -s`)

Currently looking at `rados.py` and the `run_in_thread` function looks like a good candidate to start with:

```
def run_in_thread(target, args, timeout=0):
    interrupt = False

    countdown = timeout
    t = RadosThread(target, args)

    # allow the main thread to exit (presumably, avoid a join() on this
    # subthread) before this thread terminates. This allows SIGINT
    # exit of a blocked call. See below.
    t.daemon = True

    t.start()
    try:
        # poll for thread exit
        while t.is_alive():
            t.join(POLL_TIME_INCR)
            if timeout and t.is_alive():
                countdown = countdown - POLL_TIME_INCR
                if countdown <= 0:
                    raise KeyboardInterrupt

        t.join() # in case t exits before reaching the join() above
    except KeyboardInterrupt:
        # ..but allow SIGINT to terminate the waiting. Note: this
        # relies on the Linux kernel behavior of delivering the signal
        # to the main thread in preference to any subthread (all that's
        # strictly guaranteed is that *some* thread that has the signal
        # unblocked will receive it). But there doesn't seem to be
        # any interface to create t with SIGINT blocked.
        interrupt = True

    if interrupt:
        t.retval = -errno.EINTR
```

```
return t.retval
```

#8 - 08/05/2014 10:43 PM - Dmitry Smirnov

For a moment Python maintainer in Debian kindly fixed this issue for us by adding patch to revert problematic change in Python. However this is a time bomb as it is (potentially) affecting Ceph on all architectures and distributions outside Debian. Please follow-up with Python developers or make changes for compatibility with Python-2.7.8. This is a very serious issue because no cluster components can be started with vanilla Python-2.7.8.

#9 - 10/30/2014 12:53 AM - Boris Ranto

Just a note that people are hitting this in fedora 21, now:

https://bugzilla.redhat.com/show_bug.cgi?id=1155335

#10 - 11/19/2014 06:20 PM - Dan Mick

This works around the problem, while also destroying the exit code from the ceph program, so if you rely on that, this won't help, but it will at least let the command exit:

change the last line of ceph, 'sys.exit(main())', to

```
main()
os.kill(os.getpid(), 15)
```

#11 - 11/20/2014 11:41 AM - Joe Julian

In order to get the exit code, I tried this:

```
result = main()
del cluster_handle
sys.exit(result)
```

Which resulted in a core dump:

```
Illegal instruction (core dumped)
```

Which, of course, was the same result if I tried `cluster_handle.shutdown()` (since `Rados.__del__` does the same call).

I was able to work around this by removing the `Rados.__del__` function from `rados.py`. This allowed the thread to be at least abandoned with the above code and `sys.exit` to conclude.

#12 - 12/02/2014 01:19 PM - Joe Julian

The SIGILL was cured in master with the application of 92615ea and cf2104d. I've tested backporting these to firefly which allowed shutdown to be called without crashing.

Still need to del cluster_handle to avoid the hang which I think is because it's not guaranteed that threads will garbage collect with sys.exit.

PR [#3053](#)

#13 - 12/08/2014 11:47 AM - Dan Mick

I think the right fix for this is to remove Rados.__del__. I'll come up with a pull request unless you want to, Joe.

#14 - 12/08/2014 11:48 AM - Dan Mick

- Assignee set to Dan Mick

#15 - 12/08/2014 09:44 PM - Dan Mick

- Backport set to firefly

#16 - 12/10/2014 03:21 PM - Loic Dachary

- Status changed from New to Pending Backport

giant also ?

#17 - 01/06/2015 03:17 AM - Loic Dachary

- Backport changed from firefly to firefly,giant

#18 - 02/10/2015 10:05 PM - Loic Dachary

- Status changed from Pending Backport to Resolved

#19 - 02/24/2015 02:03 PM - Loic Dachary

- merged in master with <https://github.com/ceph/ceph/pull/3119>

#20 - 02/24/2015 02:05 PM - Loic Dachary

- merged in giant by <https://github.com/ceph/ceph/pull/3168>

#21 - 02/24/2015 02:07 PM - Loic Dachary

- Status changed from Resolved to Pending Backport

#22 - 03/26/2015 03:58 PM - Loic Dachary

[e00270b](#) rados.py: remove Rados.__del__(); it just causes problems (in firefly), [ed8c9af](#) rados.py: remove Rados.__del__(); it just causes problems (in giant),

#23 - 03/26/2015 03:58 PM - Loic Dachary

- Status changed from Pending Backport to Resolved