rbd - Bug #3737

Higher ping-latency observed in qemu with rbd_cache=true during disk-write

01/07/2013 06:30 AM - Oliver Francke

Status:	Resolved	**% Done:**	0%
Priority:	High	**Spent time:**	0.00 hour
Assignee:	Josh Durgin		
Category:			
Target version:	v0.61 - Cuttlefish		
Source:	Development	**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:	bobtail	**ceph-qa-suite:**	
Regression:	No	**Pull request ID:**	
Severity:		**Crash signature:**	

Description

Hi Josh,

as per our short conversation in IRC-#ceph there is an issue with latency/responsiveness with rbd_cache enabled, no matter what cache= says.
In the lab we have qemu-1.2.2 currently as well as ceph version 0.56-111-ga14a36e (a14a36ed78d9febb7fbf1f6bf209d9bd58daace6)

Please advise necessary debug-switches to narrow down the problem.

Thnx and have a pretty good year to all of you ;)

Oliver.

Related issues:

Related to rbd - Subtask #4091: ObjectCacher: optionally make readx/writex ca...	**Resolved**	**02/11/2013**

History

#1 - 01/10/2013 09:50 AM - Sage Weil

- Priority changed from Normal to High

#2 - 01/10/2013 12:52 PM - Ian Colle

- Project changed from Ceph to rbd

- Category deleted (qemu)

- Target version deleted (v0.56)

#3 - 01/21/2013 09:35 AM - Oliver Francke

- File 905_test.log.xz added

- File ping.log.xz added

Hi Josh,

according to our conversation I did some testing.
I started the dd if=/dev... of=/tmp/doof.dat bs=4k count=256000 at around 18:10:00 as you can assume with my ping.log.
I think highest RTT was 500ms. And all above let's say 3-5ms I do not see with rbd_cache=false.

Best regards,

Oliver.

#4 - 02/20/2013 11:18 PM - Chris Dunlop

Confirmed here, with ceph-0.56.3 and qemu-1.3.1.

See attached test output.

A summary is, the average ping time, and the standard deviation of the same, is much worse with rbd_cache=1:

rbd_cache=0: Avg: 0.493 ms Std: 0.109 ms
rbd_cache=1: Avg: 148.107 ms Std: 219.786 ms

#5 - 02/20/2013 11:20 PM - Chris Dunlop

- File test.log added

Sigh. The attachment might help...

#6 - 02/21/2013 02:50 PM - Josh Durgin

I've looked at the logs, and I think #4091 should fix this. The high ping times tend to occur around when the cache fills up, making aio_write() block.

#7 - 03/01/2013 11:35 AM - Sage Weil

- Tracker changed from Bug to Fix

#8 - 03/01/2013 11:36 AM - Ian Colle

- Target version set to v0.60

#9 - 03/01/2013 11:40 AM - Sage Weil

- translation missing: en.field_story_points set to 8.00

#10 - 03/11/2013 04:53 PM - Neil Levine

- Status changed from New to 12

#11 - 03/15/2013 11:17 AM - Sage Weil

- Status changed from 12 to 7

#12 - 03/18/2013 11:24 AM - Ian Colle

- Target version changed from v0.60 to v0.61 - Cuttlefish

#13 - 03/26/2013 01:04 AM - Josh Durgin

Looks like I finally found a fix - using an explicitly asynchronous flush (instead of the sync flush made async by qemu coroutines) fixes the problem in my environment. The rest of the I/O through qemu already uses explicitly async calls, so it's something about the interaction with coroutines or the way in which qemu uses coroutines to make the sync flush async. I'd still like to dig deeper to see what the underlying issue is, and see whether it's a generic problem in qemu or a known bad idea to mix aio and qemu coroutines.

#14 - 03/26/2013 12:52 PM - Josh Durgin

There's no way around it - we need an async flush in librbd. Using coroutines vs callbacks doesn't matter in this case, if the flush is not async, there's no way for the coroutine to yield.

#15 - 03/29/2013 11:16 AM - Josh Durgin

- Status changed from 7 to Fix Under Review

#16 - 03/29/2013 01:29 PM - Josh Durgin

- Status changed from Fix Under Review to Resolved

commit:95c4a81be1af193786d0483fcbe81104d3da7c40 Note that the qemu patch still needs to get merged upstream (#4581).

#17 - 03/29/2013 01:42 PM - Josh Durgin

- Tracker changed from Fix to Bug

- Status changed from Resolved to Pending Backport

- Backport set to bobtail

#18 - 03/31/2013 01:10 PM - Stefan Priebe

Thanks for your great work! Is there already a way / branch to test this with bobtail?

#19 - 04/15/2013 12:16 AM - Josh Durgin

- Status changed from Pending Backport to 7

The branch wip-bobtail-rbd-backports-req-order has the fix for this plus several other bugs backported on top of the current bobtail branch. It passes simple testing, and is going through more thorough testing overnight.

#20 - 04/16/2013 05:50 AM - Oliver Francke

Hi Josh,

sounds promising, unfortunately I'm currently on 0.60... in our lab. We are going to move forward to latest bobtail next week in our productive env perhaps, do you think it will make it into this package?

Thnx n best regards,

Oliver.

#21 - 04/16/2013 10:46 AM - Josh Durgin

Yeah, the backports should definitely be merged by next week. On your lab cluster, you could try librbd from the 'next' branch, which has the librbd side of the fix for this.

#22 - 04/17/2013 01:00 AM - Oliver Francke

Well,

could it be, that the fix already made it into "ceph version 0.60 (f26f7a39021dbf440c28d6375222e21c94fe8e5c)"? I did not see any high latencies while writing...

Oliver.

#23 - 04/22/2013 03:02 AM - Oliver Francke

Ooops, sorry...,

was a bit misleaded, cause "cache=writeback" was still in the config file.

Oliver.

#24 - 04/23/2013 07:09 AM - Wido den Hollander

I just tested the Qemu patch with a cherry-pick to Qemu 1.2 and with the wip-bobtail-rbd-backports-req-order branch and that does indeed seem to improve the write performance a lot.

I saw about a 90% performance increase on this particular system.

#25 - 04/23/2013 12:07 PM - Josh Durgin

- Status changed from 7 to Resolved

Thanks for testing it out everyone. It's now in the bobtail branch too.

#26 - 05/25/2013 12:24 AM - Edwin Peer

Using ceph 0.61.2 and qemu 1.4.2 or earlier versions with the patch:

The following hangs after a few iterations:

```
phobos ~ # i=0; while [ $i -lt 30 ]; do dd if=/dev/zero of=test bs=4k count=1000000 conv=fdatasync; i=$[$i+1]; done
1000000+0 records in
1000000+0 records out
4096000000 bytes (4.1 GB) copied, 141.949 s, 28.9 MB/s
1000000+0 records in
1000000+0 records out
4096000000 bytes (4.1 GB) copied, 115.936 s, 35.3 MB/s
```

If I revert the qemu patch, then it no longer locks up, but the latency issue is present (even with caching disabled).

Any ideas?

#27 - 05/25/2013 07:27 AM - Edwin Peer

Update: seems to work fine if I turn writeback caching back on again (previously turned off before patching).

Files

905_test.log.xz	2.27 MB	01/21/2013	Oliver Francke
ping.log.xz	1.14 KB	01/21/2013	Oliver Francke
test.log	15.6 KB	02/20/2013	Chris Dunlop