bluestore - Bug #38559

50-100% iops lost due to bluefs_preextend_wal_files = false

03/03/2019 08:07 PM - Vitaliy Filippov

Status:	Resolved	**Start date:**	03/03/2019
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	v13.2.4, v13.2.5, v14.0.0
Backport:	mimic, nautilus, luminous	**ceph-qa-suite:**	
Regression:	No	**Pull request ID:**	26909
Severity:	1 - critical	**Crash signature:**	

Description

Hi.

I was investigating why RocksDB performance is so bad considering random 4K iops. I was looking at strace and one thing catched my eye - that was the OSD doing TWO transactions for each small (deferred) incoming write. In both mimic and nautilus strace looks like the following:

There are groups of 5 operations repeated by `bstore_kv_sync`:

- pwritev(8-12 kb, offset=1440402997248), offsets always increase
- sync_file_range(just written 8-12 kb)
- fdatasync()
- io_submit(op=pwritev, iov_len=4 kb, aio_offset=1455403167744) - offsets differ from first step, but also increase by 4 kb with each write
- fdatasync()

After every 64 such groups there come some io_submit's from `bstore_kv_final` - this is obviously the application of deferred writes, and the first pwritev is obviously the RocksDB WAL.

But what's the remaining io_submit?

It is the BlueFS's WAL! And all that it seems to do is to increase the size of RocksDB WAL and changing its modification time. So again you have "journaling of the journal"-like issue, as in old days with filestore :)

Then I found the "bluefs_preextend_wal_files" option, and yes, it disables this behaviour when set to true, and random IOPS increase by +50..+100% depending on the workload. But it corrupts the RocksDB when the OSD is shut down uncleanly. It's https://tracker.ceph.com/issues/18338 which I easily reproduced by starting a single OSD locally and writing with "fio -ioengine=rbd -direct=1 -invalidate=1 -name=test -bs=4M -iodepth=16 -rw=write -pool=bench -rbdname=testimg" into it.

I think this is REALLY ugly. Bluestore is just wasting 1/3 to 1/2 random iops performance. It must be fixed :)

Related issues:

Copied to bluestore - Backport #40280: mimic: 50-100% iops lost due to bluefs...	**Resolved**
Copied to bluestore - Backport #40281: nautilus: 50-100% iops lost due to blu...	**Resolved**
Copied to bluestore - Backport #41510: luminous: 50-100% iops lost due to blu...	**Resolved**

History

#1 - 03/04/2019 10:04 PM - Greg Farnum

- Project changed from Ceph to bluestore

#2 - 03/12/2019 02:59 PM - Sage Weil

This goes away after you write more metadta into rocksdb and it starts overwriting previous wal files. The purpose of this option is only to make things fast out of the gate for the purposes of convenient benchmarking etc.

#3 - 03/12/2019 03:00 PM - Sage Weil

- Status changed from New to Verified

#4 - 03/12/2019 03:39 PM - Vitaliy Filippov

Yes, I've thought of that but I haven't tested it... However this is rather strange then. Who does the fsync if BlueFS isn't writing to WALs? sync_file_range is called with unsafe flags so it's something else...

In fact I've submitted a pull request here https://github.com/ceph/ceph/pull/26870

It adds more wait flags to sync_file_range, makes it safe and thus makes it possible to enable bluefs_preextend_wal_files. I experience +100% iops on HDD setups with this patch and bluefs_preextend_wal_files set to true.

#5 - 03/28/2019 02:22 PM - Neha Ojha

- Status changed from Verified to Need Review

https://github.com/ceph/ceph/pull/26909

#6 - 06/11/2019 05:52 AM - Kefu Chai

- Status changed from Need Review to Pending Backport

- Backport set to mimic, nautilus

- Pull request ID set to 26909

#7 - 06/11/2019 08:19 PM - Nathan Cutler

- Copied to Backport #40280: mimic: 50-100% iops lost due to bluefs_preextend_wal_files = false added

#8 - 06/11/2019 08:19 PM - Nathan Cutler

- Copied to Backport #40281: nautilus: 50-100% iops lost due to bluefs_preextend_wal_files = false added

#9 - 08/09/2019 04:01 AM - Konstantin Shalygin

luminous: https://github.com/ceph/ceph/pull/29564

#10 - 08/26/2019 03:06 PM - Nathan Cutler

- Status changed from Pending Backport to Resolved

Having been run with --resolve-parent, the script "backport-create-issue" set the status of this issue to "Resolved" because it determined all backport issues are in status Resolved.

#11 - 08/26/2019 03:08 PM - Nathan Cutler

- Backport changed from mimic, nautilus to mimic, nautilus, luminous

#12 - 08/26/2019 03:09 PM - Nathan Cutler

- Status changed from Resolved to Pending Backport

- Target version deleted (v14.0.0)

#13 - 08/26/2019 03:09 PM - Nathan Cutler

- Copied to Backport #41510: luminous: 50-100% iops lost due to bluefs_preextend_wal_files = false added

#14 - 10/17/2019 08:14 AM - Nathan Cutler

- Status changed from Pending Backport to Resolved

While running with --resolve-parent, the script "backport-create-issue" noticed that all backports of this issue are in status "Resolved" or "Rejected".