Ceph - Bug #743

osd: broken ordering when pg ops are requeued

01/26/2011 10:03 AM - Sage Weil

Status:	Resolved	**% Done:**	0%
Priority:	Normal	**Spent time:**	2.00 hours
Assignee:	Sage Weil		
Category:	OSD		
Target version:	v0.24.3		
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:		**ceph-qa-suite:**	
Regression:	No	**Pull request ID:**	
Severity:	3 - minor	**Crash signature:**	

Description

Incoming messages race with requeued ops and get out of order. This is problematic for osd_sub_op's in partricular.

```
osd/PG.cc: In function 'void PG::add_log_entry(PG::Log::Entry&, ceph::bufferlist&)':
osd/PG.cc:2349: FAILED assert(e.version > info.last_update)
 ceph version 0.24.2 (commit:9fdaa1370ff0885ff1dee0466ce553abb08e3c03)
 1: (PG::add_log_entry(PG::Log::Entry&, ceph::buffer::list&)+0x44c) [0x5456dc]
 2: (ReplicatedPG::log_op(std::vector<PG::Log::Entry, std::allocator<PG::Log::Entry> >&, eversion_
t, ObjectStore::Transaction&)+0x56) [0x4840a6]
 3: (ReplicatedPG::sub_op_modify(MOSDSubOp*)+0xaf7) [0x4884e7]
 4: (OSD::dequeue_op(PG*)+0x304) [0x4d6544]
 5: (ThreadPool::worker()+0x291) [0x5dbbf1]
 6: (ThreadPool::WorkThread::entry()+0xd) [0x50857d]
 7: (Thread::_entry_func(void*)+0xa) [0x478eaa]
 8: /lib/libpthread.so.0 [0x7fb0c0b0f73a]
 9: (clone()+0x6d) [0x7fb0bf7b169d]
 NOTE: a copy of the executable, or `objdump -rdS <executable>` is needed to interpret this.
*** Caught signal (Aborted) ***
in thread 7fb0b53b8910
 ceph version 0.24.2 (commit:9fdaa1370ff0885ff1dee0466ce553abb08e3c03)
 1: /usr/bin/cosd [0x5eb92c]
 2: /lib/libpthread.so.0 [0x7fb0c0b17990]
 3: (gsignal()+0x35) [0x7fb0bf717f45]
 4: (abort()+0x180) [0x7fb0bf71ad80]
 5: (__gnu_cxx::__verbose_terminate_handler()+0x115) [0x7fb0bff9a975]
 6: /usr/lib/libstdc++.so.6 [0x7fb0bff98da6]
 7: /usr/lib/libstdc++.so.6 [0x7fb0bff98dd3]
 8: /usr/lib/libstdc++.so.6 [0x7fb0bff98ece]
 9: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x448) [0x5dacb8]
 a: (PG::add_log_entry(PG::Log::Entry&, ceph::buffer::list&)+0x44c) [0x5456dc]
 b: (ReplicatedPG::log_op(std::vector<PG::Log::Entry, std::allocator<PG::Log::Entry> >&, eversion_
t, ObjectStore::Transaction&)+0x56) [0x4840a6]
 c: (ReplicatedPG::sub_op_modify(MOSDSubOp*)+0xaf7) [0x4884e7]
 d: (OSD::dequeue_op(PG*)+0x304) [0x4d6544]
 e: (ThreadPool::worker()+0x291) [0x5dbbf1]
 f: (ThreadPool::WorkThread::entry()+0xd) [0x50857d]
 10: (Thread::_entry_func(void*)+0xa) [0x478eaa]
 11: /lib/libpthread.so.0 [0x7fb0c0b0f73a]
 12: (clone()+0x6d) [0x7fb0bf7b169d]
```

The handled ops (note the OOO dup):

```
2011-01-25 17:13:09.135997 7f01c743a910 osd3 14754 handle_sub_op osd_sub_op(client49757.1:205285 0
.136 10001a0283e.00000000/head [] v 14753'90884 snapset=0=[]:[] snapc=0=[]) v3 epoch 14753
2011-01-25 17:13:09.384275 7f01c743a910 osd3 14754 handle_sub_op osd_sub_op(client49757.1:205660 0
.136 10001a029b6.00000000/head [] v 14754'90885 snapset=0=[]:[] snapc=0=[]) v3 epoch 14754
2011-01-25 17:13:09.417212 7f01c743a910 osd3 14754 handle_sub_op osd_sub_op(client49757.1:206134 0
.136 10001a02b90.00000000/head [] v 14754'90886 snapset=0=[]:[] snapc=0=[]) v3 epoch 14754
2011-01-25 16:00:57.449866 7fb0b6cbc910 osd3 14832 handle_sub_op osd_sub_op(client49757.1:206539 0
.136 10001a02d24.00000000/head [] v 14826'90887 snapset=0=[]:[] snapc=0=[]) v3 epoch 14826
2011-01-25 16:00:57.580135 7fb0b6cbc910 osd3 14832 handle_sub_op osd_sub_op(client49757.1:206637 0
.136 10001a02d86.00000000/head [] v 14826'90888 snapset=0=[]:[] snapc=0=[]) v3 epoch 14826
2011-01-25 16:01:05.873528 7fb0b6cbc910 osd3 14836 handle_sub_op osd_sub_op(client49757.1:207697 0
.136 10001a031ab.00000000/head [] v 14826'90889 snapset=0=[]:[] snapc=0=[]) v3 epoch 14826
2011-01-25 16:01:05.892665 7fb0b74bd910 osd3 14836 handle_sub_op osd_sub_op(client49757.1:206637 0
.136 10001a02d86.00000000/head [] v 14826'90888 snapset=0=[]:[] snapc=0=[]) v3 epoch 14826
```

Associated revisions

Revision fbcf6690 - 01/26/2011 06:08 PM - Sage Weil

osd: preserve ordering when ops are requeued

Requeue ops under osd_lock to preserve ordering wrt incoming messages.
Also drain the waiter queue when ms_dispatch takes the lock before calling
_dispatch(m).

Fixes: #743
Signed-off-by: Sage Weil <sage@newdream.net>

History

#1 - 01/26/2011 10:18 AM - Sage Weil

- Status changed from New to Resolved

fbcf66906e67adbe6769ba7b1853dd0161e977c6