From Zhiqiang Liu <>
Subject [PATCH V2] bcache: fix potential deadlock problem in btree_gc_coalesce
Date Sun, 26 Apr 2020 16:06:27 +0800

```
From: Zhiqiang Liu <liuzhiqiang26@huawei.com>

coccicheck reports:
  drivers/md//bcache/btree.c:1538:1-7: preceding lock on line 1417

btree_gc_coalesce func is designed to coalesce two adjacent nodes in
new_nodes[GC_MERGE_NODES] and finally release one node. All nodes`write_lock,
new_nodes[i]->write_lock, are holded before coalescing adjacent nodes,
and them will be released after coalescing successfully.

However, if the coalescing process fails, such as no enough space of new_nodes[1]
to fit all of the remaining keys in new_nodes[0] and realloc keylist failed, we
will goto to out_nocoalesce tag directly without releasing new_nodes[i]->write_lock.
Then, a deadlock will occur after calling btree_node_free to free new_nodes[i],
which also try to acquire new_nodes[i]->write_lock.

Here, we add a new tag 'out_unlock_nocoalesce' before out_nocoalesce tag to release
new_nodes[i]->write_lock when coalescing process fails.

--
V1->V2: rewrite commit log (suggested by Coly Li) and rename the patch

Fixes: 2a285686c1 ("bcache: btree locking rework")
Signed-off-by: Zhiqiang Liu <liuzhiqiang26@huawei.com>
---
 drivers/md/bcache/btree.c | 8 ++++++--
 1 file changed, 6 insertions(+), 2 deletions(-)

diff --git a/drivers/md/bcache/btree.c b/drivers/md/bcache/btree.c
index fa872df4e770..cad8b0b97e33 100644
--- a/drivers/md/bcache/btree.c
+++ b/drivers/md/bcache/btree.c
@@ -1447,7 +1447,7 @@ static int btree_gc_coalesce(struct btree *b, struct btree_op *op,
                        if (__set_blocks(n1, n1->keys + n2->keys,
                                         block_bytes(b->c)) >
                            btree_blocks(new_nodes[i]))
-                           goto out_nocoalesce;
+                           goto out_unlock_nocoalesce;

                        keys = n2->keys;
                        /* Take the key of the node we're getting rid of */
@@ -1476,7 +1476,7 @@ static int btree_gc_coalesce(struct btree *b, struct btree_op *op,

                if (__bch_keylist_realloc(&keylist,
                                          bkey_u64s(&new_nodes[i]->key)))
-                       goto out_nocoalesce;
+                       goto out_unlock_nocoalesce;

                bch_btree_node_write(new_nodes[i], &cl);
                bch_keylist_add(&keylist, &new_nodes[i]->key);
@@ -1522,6 +1522,10 @@ static int btree_gc_coalesce(struct btree *b, struct btree_op *op,
        /* Invalidated our iterator */
        return -EINTR;

+out_unlock_nocoalesce:
+       for (i = 0; i < nodes; i++)
+               mutex_unlock(&new_nodes[i]->write_lock);
+
 out_nocoalesce:
        closure_sync(&cl);

--
2.19.1
```