rgw - Bug #11749
rgw: rados objects wrongly deleted
05/23/2015 08:56 AM - xingyi wu

<table>
<thead>
<tr>
<th>Status:</th>
<th>Resolved</th>
<th>Start date:</th>
<th>05/23/2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td>05/23/2015</td>
</tr>
<tr>
<td>Assignee:</td>
<td>xingyi wu</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td>% Done:</td>
<td>60%</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Source:</td>
<td>Community (dev)</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Tags:</td>
<td></td>
<td>Reviewed:</td>
<td></td>
</tr>
<tr>
<td>Backport:</td>
<td>hammer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regression:</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Severity:</td>
<td>1 - critical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description
When testing rados gateway(giant v0.87), I found two bugs which would be dangerous enough to cause data corruption. The first one is fixed by Yehuda on https://github.com/ceph/ceph/pull/4661, in such condition, the first stripe of a part will be lost. Then I tested with upstream code, and found is still exists. Here is the details of the second bug.

This bug only happened with multipart is enabled. When uploading object with multiple parts, a part which is not completely uploaded will be destroyed by calling dispose_processor. This would occasionally cause race condition: the first upload would be possible to delete objects belong to the second upload, which would finally cause data corruption.

I set multipart size to 64MB and test it with s3cmd, actually I reproduced it with cyberduck, too. You can reproduce it with the following script:

dd if=/dev/zero of=BREAKDOWN bs=65M count=1
originalMD5=`md5sum ./BREAKDOWN | awk '{print $1}'`
rm $FILENAME
s3cmd put $FILENAME s3://BREAKDOWN/$FILENAME &
sleep 2
kill -9 `ps aux | grep "s3cmd put" | grep -v grep | awk '{print $2}``
originalMD5=`md5sum ./BREAKDOWN | awk '{print $1}'`
s3cmd get s3://BREAKDOWN/$FILENAME downloadedfile --force
downloadMD5=`md5sum .downloadedfile | awk '{print $1]'`
if [[ "$originalMD5" == "$downloadMD5" ]]; then
echo "bad MD5"
exit -1
fi

You can hit "bad MD5" bug after run enough times, when list the rados objects by "rados list -p .rgw.buckets | grep BREAKDOWN", you will find some rados objects would have already been deleted. Here is a sample:

[root@cephdev141 src]$ rados ls -p .rgw.buckets | grep 20150523164305 | sort
default.54105.4_20150523164305
default.54105.4__multipart_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1
default.54105.4__multipart_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.2
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_1
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_10
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_11
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_12
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_13
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_14
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_15
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_16
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_17
default.54105.4__shadow_20150523164305.2~JeapLhRaTUmiliBFOdYRiaBgOiUVo.1_18

You can hit "bad MD5" bug after run enough times, when list the rados objects by "rados list -p .rgw.buckets | grep BREAKDOWN", you will find some rados objects would have already been deleted. Here is a sample:
In this sample, `default.54105.4_shadow_20150523164305.2~JeapILhRaTUmiiLjBFOdYRiaBgO1UVo.1_3` and `default.54105.4_shadow_20150523164305.2~JeapILhRaTUmiiLjBFOdYRiaBgO1UVo.1_4` were wrongly deleted.

**Related issues:**
Copied to Backport #12099: rgw: rados objects wrongly deleted

**Resolved**
05/23/2015 05/23/2015

**Associated revisions**

**Revision ac1e729a - 06/10/2015 06:58 AM - xingyi wu**
rgw: fix data corruption when race condition

We should delete the object in the multipart namespace lastly to prevent a previous upload wrongly deleting objects belong to the following upload.

Fixes: #11749
Signed-off-by: wuxingyi <wuxingyi@letv.com>

**Revision 8abc46a1 - 07/01/2015 04:43 PM - xingyi wu**
rgw: fix data corruption when race condition

We should delete the object in the multipart namespace lastly to prevent a previous upload wrongly deleting objects belong to the following upload.

Fixes: #11749
Signed-off-by: wuxingyi <wuxingyi@letv.com>
(cherry picked from commit ac1e729a75b5d995028bbc223bcf5ecce0d112cc)

**History**

#1 - 05/23/2015 04:05 PM - Kefu Chai
- Status changed from New to Need Review
- Source changed from Development to Community (dev)

https://github.com/ceph/ceph/pull/4752

#2 - 06/12/2015 06:24 PM - Yehuda Sadeh
- Status changed from Need Review to Pending Backport
- Backport set to hammer

Merged in commit:69989ffa3cabe209404504edd24bd2d2a59e33e15.
backporting to firefly will also require picking up fix for #10311.

#3 - 07/01/2015 05:08 PM - Abhishek Lekshmanan
Hammer backport: https://github.com/ceph/ceph/pull/5117
#4 - 07/07/2015 10:42 PM - Gleb Borisov
Is there any way to find objects affected by this issue in bucket?

#5 - 09/02/2015 10:42 AM - Loic Dachary
- Status changed from Pending Backport to Resolved