rgw - Bug #54124

smaller object workloads wane and terminate after a few hours

02/03/2022 02:04 PM - Tim Wilkinson

Status:	Resolved	**% Done:**	0%
Priority:	High	**Spent time:**	0.00 hour
Assignee:	Mark Kogan		
Category:			
Target version:			
Source:		**Affected Versions:**	v17.0.0
Tags:		**ceph-qa-suite:**	rados
Backport:		**Pull request ID:**	
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	
Reviewed:			

Description

In our initial baseline tests with quincy, we use two sets of object sizes.

The generic sized workload looks like this ...

- 50% uses 1KB objs
- 15% 64KB objs
- 15% 8MB objs
- 15% 64MB objs
- 5% 1024MB objs

... while the smaller sized workload uses ...

- 25% between 1KB and 2KB objs
- 40% between 2KB and 4KB objs
- 25% between 4KB and 8KB objs
- 10% between 8KB and 256KB objs

The generic sized workload has executed without issues throughout the full COSBench test cycle ... # cluster fill (prepare) # 1hr hybrid measurement (44% read, 36% write, 15% list, 5% delete) # 48hr hybrid aging run # 1hr hybrid measurement # 2hr delete-write run

... but the smaller object workload does not seem to be able to maintain IO beyond a few hours. The IO wanes until zero and the job self terminates prematurely (see graph attached).

We've attempted a few things to see what may have an impact ... * using –bulk option when creating the data pool * executing the tests entirely from within the cephadm shell * presharding all buckets * reducing COSbench worker counts (which allowed the 1hr hybrid job to execute for the duration) * hardwiring the index & data pool PGs and disabling PG autoscaler

but none have had any effect at allowing the extended runs to maintain IO beyond 3 to 4 hrs.

We see nothing in MON or RGW logs that shed any light on the issue as yet.

History

#1 - 02/03/2022 02:10 PM - Tim Wilkinson

An example of the workload, this the 48hr hybrid test ...

```
    <workstage name="MAIN">
         <work name="hybridSS" workers="50" runtime="172800" >
        <operation name="readOP" type="read" ratio="44" config="cprefix=bucket;oprefix=secondary;containers=u(
4,6);objects=u(1,300000);hashCheck=true" />
            <operation name="listOP" type="list" ratio="15" config="cprefix=bucket;oprefix=secondary;container
s=u(4,6);objects=u(1,300000);hashCheck=true" />
```

```
            <operation name="writeOP" type="write" ratio="25" config="cprefix=bucket;oprefix=secondary;containers=u(1,3);objects=u(1,300000);sizes=h(1|1|50,64|64|15,8192|8192|15,65536|65536|15,1048576|1048576|5)KB"/>
            <operation name="deleteOP" type="delete" ratio="16" config="cprefix=bucket;oprefix=secondary;containers=u(1,3);objects=u(1,300000)" />
        </work>
    </workstage>
```

#2 - 02/03/2022 02:12 PM - Tim Wilkinson

Correction, the above .xml is the generic sized workload that executes without issue.

Below is the workload in question that does not succeed.

```
    <workstage name="MAIN">
      <!--  <work name="hybridSS-48hr" workers="2100" runtime="172800" >   -->
        <work name="hybridSS-48hr" workers="2100" runtime="36000" >
        <operation name="readOP" type="read" ratio="44" config="cprefix=bucket;oprefix=secondary;containers=u(4,6);objects=u(1,50000000);hashCheck=true" />
            <operation name="listOP" type="list" ratio="15" config="cprefix=bucket;oprefix=secondary;containers=u(4,6);objects=u(1,50000000);hashCheck=true" />
            <operation name="writeOP" type="write" ratio="36" config="cprefix=bucket;oprefix=secondary;containers=u(1,3);objects=u(1,50000000);sizes=h(1|2|25,2|4|40,4|8|25,8|256|10)KB" />
            <operation name="deleteOP" type="delete" ratio="5" config="cprefix=bucket;oprefix=secondary;containers=u(1,3);objects=u(1,50000000)" />
        </work>
    </workstage>
```

#3 - 02/03/2022 02:16 PM - Tim Wilkinson

```
# ceph versions
{
    "mon": {
        "ceph version 17.0.0-10292-gd34c9171 (d34c917198765ac4750818b2bf334e91a0d6b085) quincy (dev)": 3
    },
    "mgr": {
        "ceph version 17.0.0-10292-gd34c9171 (d34c917198765ac4750818b2bf334e91a0d6b085) quincy (dev)": 3
    },
    "osd": {
        "ceph version 17.0.0-10292-gd34c9171 (d34c917198765ac4750818b2bf334e91a0d6b085) quincy (dev)": 192
    },
    "mds": {},
    "rgw": {
        "ceph version 17.0.0-10292-gd34c9171 (d34c917198765ac4750818b2bf334e91a0d6b085) quincy (dev)": 8
    },
    "overall": {
        "ceph version 17.0.0-10292-gd34c9171 (d34c917198765ac4750818b2bf334e91a0d6b085) quincy (dev)": 206
    }
}
```

#4 - 02/03/2022 03:16 PM - Casey Bodley

- Assignee set to Mark Kogan

#5 - 02/03/2022 04:14 PM - Casey Bodley

- Priority changed from Normal to Urgent

#6 - 02/03/2022 04:25 PM - Vikhyat Umrao

> The generic sized workload has executed without issues throughout the full COSBench test cycle ..

Tim - can you please provide the generic size workload histogram list.

#7 - 02/03/2022 04:37 PM - Vikhyat Umrao

Vikhyat Umrao wrote:

> > The generic sized workload has executed without issues throughout the full COSBench test cycle ..
>
> Tim - can you please provide the generic size workload histogram list.

ahh I see it mentioned in note-1.

```
sizes=h(1|1|50,64|64|15,8192|8192|15,65536|65536|15,1048576|1048576|5)KB"/>
```

#8 - 02/04/2022 06:44 PM - Tim Wilkinson

rgw and rados bench workloads were executed without apparent issue. Using s3cmd to create buckets, populate them, list them, and delete them also succeeded without issue. Further searches of more recent RGW logs have shown some errors but most appear to be reshard related.

e.g.,

```
2022-02-01T18:58:45.711+0000 7f66c923b700  1 ====== starting new request req=0x7f665525d650 =====2022-02-01T18
:58:45.711+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initialize BucketShar
```

d, obj=bucket1:primary1518615 r=-22022-02-01T18:58:45.711+0000 7f675c361700 0 rgw index completion thread: ER
ROR: process(): failed to initialize BucketShard, obj=bucket1:primary14494969 r=-22022-02-01T18:58:45.711+0000
 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to initialize BucketShard, obj=bucket1:
primary14524521 r=-22022-02-01T18:58:45.711+0000 7f675c361700 0 rgw index completion thread: ERROR: process()
: failed to initialize BucketShard, obj=bucket1:primary6072224 r=-22022-02-01T18:58:45.711+0000 7f666196c700
0 req 59654201749794763 0.002000032s s3:put_obj NOTICE: resharding operation on bucket index detected, blockin
g2022-02-01T18:58:45.711+0000 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=b

2022-02-01T18:59:30.731+0000 7f675c361700 0 rgw index completion thread: int RGWRados::block_while_resharding
(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider
*) ERROR: bucket is still resharding, please retry2022-02-01T18:59:30.731+0000 7f666196c700 0 req 59654201749
794763 45.021717072s s3:put_obj int RGWRados::block_while_resharding(RGWRados::BucketShard*, std::__cxx11::str
ing*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider*) ERROR: bucket is still resharding, plea
se retry2022-02-01T18:59:30.731+0000 7f66bd223700 0 req 13172718335458426190 45.019718170s s3:put_obj int RGW
Rados::block_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yi
eld, const DoutPrefixProvider*) ERROR: bucket is still resharding, please retry2022-02-01T18:59:30.732+0000 7f
675c361700 0 rgw index completion thread: NOTICE: resharding operation on bucket index detected, blocking2022
-02-01T18:59:30.733+0000 7f673cb22700 0 req 13172718335458426190 45.021717072s s3:put_obj NOTICE: resharding
 operation on bucket index detected, blocking2022-02-01T18:59:30.734+0000 7f6735313700 0 req 59654201749794763
 45.024719238s s3:put_obj NOTICE: resharding operation on bucket index detected, blocking2022-02-01T18:59:30.7
34+0000 7f675c361700 0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-4885-8f7a-450dafefe238.
58706.1 to be held by another RGW process; skipping for now2022-02-01T18:59:30.734+0000 7f670f2c7700 0 req 17
181042643754048548 45.022716522s s3:put_obj int RGWRados::block_while_resharding(RGWRados::BucketShard*, std::
__cxx11::string*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider*) ERROR: bucket is still resh
arding, please retry2022-02-01T18:59:30.735+0000 7f673cb22700 0 INFO: RGWReshardLock::lock found lock on buck
et1:0dd08fbf-f19b-4885-8f7a-450dafefe238.58706.1 to be held by another RGW process; skipping for now2022-02-01
T18:59:30.735+0000 7f6735313700 0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-4885-8f7a-45
0dafefe238.58706.1 to be held by another RGW process; skipping for now2022-02-01T18:59:30.735+0000 7f667c1a170
0 0 req 17181042643754048548 45.023715973s s3:put_obj NOTICE: resharding operation on bucket index detected,
blocking2022-02-01T18:59:30.736+0000 7f667c1a1700 0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf
-f19b-4885-8f7a-450dafefe238.58706.1 to be held by another RGW process; skipping for now2022-02-01T18:59:30.73
6+0000 7f6719adc700 0 req 84868893635189860 87 45.021717072s s3:put_obj int RGWRados::block_while_resharding(R
GWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider*)
 ERROR: bucket is still resharding, please retry2022-02-01T18:59:30.737+0000 7f66b5a14700 0 req 67539789053675
677197 45.024719238s s3:put_obj int RGWRados::block_while_resharding(RGWRados::BucketShard*, std::__cxx11::str
ing*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider*) ERROR: bucket is still resharding, plea
se retry

2022-02-01T19:08:42.311+0000 7f6708aba700 1 ====== starting new request req=0x7f665813a650 =====2022-02-01T19
:08:42.311+0000 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to initialize BucketShar
d, obj=bucket1:primary1075470 r=-22022-02-01T19:08:42.311+0000 7f675c361700 0 rgw index completion thread: ER
ROR: process(): failed to initialize BucketShard, obj=bucket1:primary2980378 r=-22022-02-01T19:08:42.311+0000
 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to initialize BucketShard, obj=bucket1:p
rimary12980484 r=-22022-02-01T19:08:42.311+0000 7f675c361700 0 rgw index completion thread: ERROR: process():
 failed to initialize BucketShard, obj=bucket1:primary36491238 r=-22022-02-01T19:08:42.311+0000 7f675c361700
0 rgw index completion thread: ERROR: process(): failed to initialize BucketShard, obj=bucket1:primary38039779
 r=-22022-02-01T19:08:42.311+0000 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to ini
tialize BucketShard, obj=bucket1:primary8455985 r=-22022-02-01T19:08:42.311+0000 7f675c361700 0 rgw index com
pletion thread: ERROR: process(): failed to initialize BucketShard, obj=bucket1:primary33575227 r=-22022-02-01
T19:08:42.311+0000 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to initialize BucketS
hard, obj=bucket1:primary11402798 r=-22022-02-01T19:08:42.311+0000 7f675c361700 0 rgw index completion thread
: ERROR: process(): failed to initialize BucketShard, obj=bucket1:primary19795781 r=-22022-02-01T19:08:42.311+
0000 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to initialize BucketShard, obj=buck
et1:primary37116996 r=-22022-02-01T19:08:42.311+0000 7f675c361700 0 rgw index completion thread: ERROR: proce
ss(): failed to initialize BucketShard, obj=bucket1:primary23664237 r=-22022-02-01T19:08:42.311+0000 7f675c361
700 0 rgw index completion thread: ERROR: process(): failed to initialize BucketShard, obj=bucket1:primary485
5062 r=-22022-02-01T19:08:42.311+0000 7f675c361700 0 rgw index completion thread: ERROR: process(): failed to
 initialize BucketShard, obj=bucket1:primary49141164 r=-2

2022-02-01T19:09:22.630+0000 7f6714ad2700 0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58721.1 to be held by another RGW process; skipping for now2022-02-01T19:09:22.670+0000 7
f6663970700 0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-4885-8f7a-450dafefe238.58721.1 t
o be held by another RGW process; skipping for now2022-02-01T19:09:22.674+0000 7f670eac6700 0 INFO: RGWReshar
dLock::lock found lock on bucket1:0dd08fbf-f19b-4885-8f7a-450dafefe238.58721.1 to be held by another RGW proce
ss; skipping for now2022-02-01T19:09:22.754+0000 7f66f7a98700 0 INFO: RGWReshardLock::lock found lock on buck
et1:0dd08fbf-f19b-4885-8f7a-450dafefe238.58721.1 to be held by another RGW process; skipping for now2022-02-01
T19:09:27.345+0000 7f672aafe700 0 req 16881230217162373146 45.036758423s s3:put_obj int RGWRados::block_while
_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, const DoutPre
fixProvider*) ERROR: bucket is still resharding, please retry2022-02-01T19:09:27.347+0000 7f6675994700 0 req
14346839619327919021 45.037757874s s3:put_obj int RGWRados::block_while_resharding(RGWRados::BucketShard*, std
::__cxx11::string*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider*) ERROR: bucket is still re
sharding, please retry2022-02-01T19:09:27.347+0000 7f66f2a8e700 0 req 16881230217162373146 45.038757324s s3:p
ut_obj NOTICE: resharding operation on bucket index detected, blocking2022-02-01T19:09:27.347+0000 7f674c34170
0 0 req 1523280096868581 42 45.035758972s s3:put_obj int RGWRados::block_while_resharding(RGWRados::BucketShar

```
d*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider*) ERROR: bucket is s
till resharding, please retry

2022-02-02T02:30:10.586+0000 7f66e7277700  1 ====== starting new request req=0x7f6645d77650 =====2022-02-02T02
:30:10.586+0000 7f66e7277700  0 req 16883362725914845201 0.000000000s s3:get_obj Scheduling request failed wit
h -22182022-02-02T02:30:10.586+0000 7f66e7277700  1 req 16883362725914845201 0.000000000s op->ERRORHANDLER: er
r_no=-2218 new_err_no=-22182022-02-02T02:30:10.586+0000 7f66e7277700  1 ====== req done req=0x7f6645d77650 op
status=0 http_status=503 latency=0.000000000s ======2022-02-02T02:30:10.586+0000 7f66e7277700  1 beast: 0x7f66
45d77650: 172.19.45.16 - - [02/Feb/2022:02:30:10.586 +0000] "GET /bucket5/primary29690260 HTTP/1.1" 503 181 -
"aws-sdk-java/1.10.76 Linux/4.18.0-348.2.1.el8_5.x86_64 OpenJDK_64-Bit_Server_VM/25.312-b07/1.8.0_312" - laten
cy=0.000000000s2022-02-02T02:30:10.587+0000 7f6734b12700  1 ====== starting new request req=0x7f6645d77650 ===
==2022-02-02T02:30:10.587+0000 7f6734b12700  0 req 4343398786817728859 0.000000000s s3:get_obj Scheduling requ
est failed with -22182022-02-02T02:30:10.587+0000 7f6734b12700  1 req 4343398786817728859 0.000000000s op->ERR
ORHANDLER: err_no=-2218 new_err_no=-22182022-02-02T02:30:10.587+0000 7f66d5253700  1 ====== starting new reque
st req=0x7f6642d98650 =====2022-02-02T02:30:10.587+0000 7f673230d700  1 ====== req done req=0x7f663ad19650 op
status=0 http_status=200 latency=0.053000849s ======2022-02-02T02:30:10.587+0000 7f66d5253700  0 req 549486086
9981029002 0.000000000s s3:get_obj Scheduling request failed with -22182022-02-02T02:30:10.587+0000 7f66d52537
00  1 req 5494860869981029002 0.000000000s op->ERRORHANDLER: err_no=-2218 new_err_no=-22182022-02-02T02:30:10.
587+0000 7f673230d700  1 beast: 0x7f663ad19650: 172.19.44.88 - johndoe [02/Feb/2022:02:30:10.533 +0000] "GET /
bucket5/primary40397729 HTTP/1.1" 200 2000 - "aws-sdk-java/1.10.76 Linux/4.18.0-193.el8.x86_64 OpenJDK_64-Bit_
Server_VM/25.312-b07/1.8.0_312" - latency=0.053000849s2022-02-02T02:30:10.587+0000 7f6757b58700  1 ====== req
done req=0x7f6645d77650 op status=0 http_status=503 latency=0.000000000s ======
```

#9 - 02/04/2022 06:50 PM - Tim Wilkinson

Let's try that again with some newlines ...

```
2022-02-01T18:58:45.711+0000 7f66c923b700  1 ====== starting new request req=0x7f665525d650 =====
2022-02-01T18:58:45.711+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initiali
ze BucketShard, obj=bucket1:primary1518615 r=-
22022-02-01T18:58:45.711+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary14494969 r=-
22022-02-01T18:58:45.711+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary14524521 r=-
22022-02-01T18:58:45.711+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary6072224 r=-
22022-02-01T18:58:45.711+0000 7f666196c700  0 req 59654201749794763 0.002000032s s3:put_obj NOTICE: resharding
 operation on bucket index detected, blocking
2022-02-01T18:58:45.711+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initiali
ze BucketShard, obj=b

2022-02-01T18:59:30.731+0000 7f675c361700  0 rgw index completion thread: int RGWRados::block_while_resharding
(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, const DoutPrefixProvider
*) ERROR: bucket is still resharding, please retry
2022-02-01T18:59:30.731+0000 7f666196c700  0 req 59654201749794763 45.021717072s s3:put_obj int RGWRados::bloc
k_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, const
DoutPrefixProvider*) ERROR: bucket is still resharding, please retry
2022-02-01T18:59:30.731+0000 7f66bd223700  0 req 13172718335458426190 45.019718170s s3:put_obj int RGWRados::b
lock_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, con
st DoutPrefixProvider*) ERROR: bucket is still resharding, please retry
2022-02-01T18:59:30.732+0000 7f675c361700  0 rgw index completion thread: NOTICE: resharding operation on buck
et index detected, blocking
```

```
2022-02-01T18:59:30.733+0000 7f673cb22700  0 req 13172718335458426190 45.021717072s s3:put_obj NOTICE: reshard
ing operation on bucket index detected, blocking
2022-02-01T18:59:30.734+0000 7f6735313700  0 req 59654201749794763 45.024719238s s3:put_obj NOTICE: resharding
 operation on bucket index detected, blocking
2022-02-01T18:59:30.734+0000 7f675c361700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58706.1 to be held by another RGW process; skipping for now
2022-02-01T18:59:30.734+0000 7f670f2c7700  0 req 17181042643754048548 45.022716522s s3:put_obj int RGWRados::b
lock_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, con
st DoutPrefixProvider*) ERROR: bucket is still resharding, please retry
2022-02-01T18:59:30.735+0000 7f673cb22700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58706.1 to be held by another RGW process; skipping for now
2022-02-01T18:59:30.735+0000 7f6735313700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58706.1 to be held by another RGW process; skipping for now
2022-02-01T18:59:30.735+0000 7f667c1a1700  0 req 17181042643754048548 45.023715973s s3:put_obj NOTICE: reshard
ing operation on bucket index detected, blocking
2022-02-01T18:59:30.736+0000 7f667c1a1700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58706.1 to be held by another RGW process; skipping for now
2022-02-01T18:59:30.736+0000 7f6719adc700  0 req 8486889363518986087 45.021717072s s3:put_obj int RGWRados::bl
ock_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, cons
t DoutPrefixProvider*) ERROR: bucket is still resharding, please retry
2022-02-01T18:59:30.737+0000 7f66b5a14700  0 req 67539789053676771 97 45.024719238s s3:put_obj int RGWRados::bl
ock_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, cons
t DoutPrefixProvider*) ERROR: bucket is still resharding, please retry

2022-02-01T19:08:42.311+0000 7f6708aba700  1 ====== starting new request req=0x7f665813a650 =====
2022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initiali
ze BucketShard, obj=bucket1:primary1075470 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary2980378 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary12980484 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary36491238 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary38039779 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary8455985 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary33575227 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary11402798 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary19795781 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary37116996 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary23664237 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary4855062 r=-
22022-02-01T19:08:42.311+0000 7f675c361700  0 rgw index completion thread: ERROR: process(): failed to initial
ize BucketShard, obj=bucket1:primary49141164 r=-2

2022-02-01T19:09:22.630+0000 7f6714ad2700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58721.1 to be held by another RGW process; skipping for now
2022-02-01T19:09:22.670+0000 7f6663970700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58721.1 to be held by another RGW process; skipping for now
2022-02-01T19:09:22.674+0000 7f670eac6700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58721.1 to be held by another RGW process; skipping for now
2022-02-01T19:09:22.754+0000 7f66f7a98700  0 INFO: RGWReshardLock::lock found lock on bucket1:0dd08fbf-f19b-48
85-8f7a-450dafefe238.58721.1 to be held by another RGW process; skipping for now
2022-02-01T19:09:27.345+0000 7f672aafe700  0 req 16881230217162373146 45.036758423s s3:put_obj int RGWRados::b
lock_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, con
st DoutPrefixProvider*) ERROR: bucket is still resharding, please retry
2022-02-01T19:09:27.347+0000 7f6675994700  0 req 14346839619327919021 45.037757874s s3:put_obj int RGWRados::b
lock_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, con
st DoutPrefixProvider*) ERROR: bucket is still resharding, please retry
2022-02-01T19:09:27.347+0000 7f66f2a8e700  0 req 16881230217162373146 45.038757324s s3:put_obj NOTICE: reshard
ing operation on bucket index detected, blocking
2022-02-01T19:09:27.347+0000 7f674c341700  0 req 152328009686858142 45.035758972s s3:put_obj int RGWRados::blo
ck_while_resharding(RGWRados::BucketShard*, std::__cxx11::string*, const RGWBucketInfo&, optional_yield, const
 DoutPrefixProvider*) ERROR: bucket is still resharding, please retry

2022-02-02T02:30:10.586+0000 7f66e7277700  1 ====== starting new request req=0x7f6645d77650 =====
2022-02-02T02:30:10.586+0000 7f66e7277700  0 req 16883362725914845201 0.000000000s s3:get_obj Scheduling reque
st failed with -2218
```

```
2022-02-02T02:30:10.586+0000 7f66e7277700  1 req 16883362725914845201 0.000000000s op->ERRORHANDLER: err_no=-2
218 new_err_no=-2218
2022-02-02T02:30:10.586+0000 7f66e7277700  1 ====== req done req=0x7f6645d77650 op status=0 http_status=503 la
tency=0.000000000s ======
2022-02-02T02:30:10.586+0000 7f66e7277700  1 beast: 0x7f6645d77650: 172.19.45.16 - - [02/Feb/2022:02:30:10.586
 +0000] "GET /bucket5/primary29690260 HTTP/1.1" 503 181 - "aws-sdk-java/1.10.76 Linux/4.18.0-348.2.1.el8_5.x86
_64 OpenJDK_64-Bit_Server_VM/25.312-b07/1.8.0_312" - latency=0.000000000s
2022-02-02T02:30:10.587+0000 7f6734b12700  1 ====== starting new request req=0x7f6645d77650 =====
2022-02-02T02:30:10.587+0000 7f6734b12700  0 req 4343398786817728859 0.000000000s s3:get_obj Scheduling reques
t failed with -2218
2022-02-02T02:30:10.587+0000 7f6734b12700  1 req 4343398786817728859 0.000000000s op->ERRORHANDLER: err_no=-22
18 new_err_no=-2218
2022-02-02T02:30:10.587+0000 7f66d5253700  1 ====== starting new request req=0x7f6642d98650 =====2022-02-02T02
:30:10.587+0000 7f673230d700  1 ====== req done req=0x7f663ad19650 op status=0 http_status=200 latency=0.05300
0849s ======
2022-02-02T02:30:10.587+0000 7f66d5253700  0 req 5494860869981029002 0.000000000s s3:get_obj Scheduling reques
t failed with -2218
2022-02-02T02:30:10.587+0000 7f66d5253700  1 req 5494860869981029002 0.000000000s op->ERRORHANDLER: err_no=-22
18 new_err_no=-2218
2022-02-02T02:30:10.587+0000 7f673230d700  1 beast: 0x7f663ad19650: 172.19.44.88 - johndoe [02/Feb/2022:02:30:
10.533 +0000] "GET /bucket5/primary40397729 HTTP/1.1" 200 2000 - "aws-sdk-java/1.10.76 Linux/4.18.0-193.el8.x8
6_64 OpenJDK_64-Bit_Server_VM/25.312-b07/1.8.0_312" - latency=0.053000849s
2022-02-02T02:30:10.587+0000 7f6757b58700  1 ====== req done req=0x7f6645d77650 op status=0 http_status=503 la
tency=0.000000000s ======
```

#10 - 02/07/2022 12:17 PM - Mark Kogan

thank you for the detailed information for analysis,

from the logs in https://tracker.ceph.com/issues/54124#note-9
the reason for the op rejection is '503' (slow down):

```
2022-02-02T02:30:10.586+0000 7f66e7277700  1 ====== starting new request req=0x7f6645d77650 =====
2022-02-02T02:30:10.586+0000 7f66e7277700  0 req 16883362725914845201 0.000000000s s3:get_obj Scheduling reque
st failed with -2218
2022-02-02T02:30:10.586+0000 7f66e7277700  1 req 16883362725914845201 0.000000000s op->ERRORHANDLER: err_no=-2
218 new_err_no=-2218
2022-02-02T02:30:10.586+0000 7f66e7277700  1 ====== req done req=0x7f6645d77650 op status=0 http_status=503 la
tency=0.000000000s ======
2022-02-02T02:30:10.586+0000 7f66e7277700  1 beast: 0x7f6645d77650: 172.19.45.16 - - [02/Feb/2022:02:30:10.586
 +0000] "GET /bucket5/primary29690260 HTTP/1.1" 503 181 - "aws-sdk-java/1.10.76 Linux/4.18.0-348.2.1.el8_5.x86
_64 OpenJDK_64-Bit_Server_VM/25.312-b07/1.8.0_312" - latency=0.000000000s
2022-02-02T02:30:10.587+0000 7f6734b12700  1 ====== starting new request req=0x7f6645d77650 =====
2022-02-02T02:30:10.587+0000 7f6734b12700  0 req 4343398786817728859 0.000000000s s3:get_obj Scheduling reques
t failed with -2218
2022-02-02T02:30:10.587+0000 7f6734b12700  1 req 4343398786817728859 0.000000000s op->ERRORHANDLER: err_no=-22
18 new_err_no=-2218
2022-02-02T02:30:10.587+0000 7f66d5253700  1 ====== starting new request req=0x7f6642d98650 =====2022-02-02T02
:30:10.587+0000 7f673230d700  1 ====== req done req=0x7f663ad19650 op status=0 http_status=200 latency=0.05300
0849s ======
```

```
2022-02-02T02:30:10.587+0000 7f66d5253700  0 req 5494860869981029002 0.000000000s s3:get_obj Scheduling reques
t failed with -2218
2022-02-02T02:30:10.587+0000 7f66d5253700  1 req 5494860869981029002 0.000000000s op->ERRORHANDLER: err_no=-22
18 new_err_no=-2218
2022-02-02T02:30:10.587+0000 7f673230d700  1 beast: 0x7f663ad19650: 172.19.44.88 - johndoe [02/Feb/2022:02:30:
10.533 +0000] "GET /bucket5/primary40397729 HTTP/1.1" 200 2000 - "aws-sdk-java/1.10.76 Linux/4.18.0-193.el8.x8
6_64 OpenJDK_64-Bit_Server_VM/25.312-b07/1.8.0_312" - latency=0.053000849s
2022-02-02T02:30:10.587+0000 7f6757b58700  1 ====== req done req=0x7f6645d77650 op status=0 http_status=503 la
tency=0.000000000s ======
                                                                                                        ^^^
                                                                                                        ^^^
```

from the workload specified in https://tracker.ceph.com/issues/54124#note-2, cosbench workers are instructed to generate 2100 concurrent requests:

```
    <workstage name="MAIN">
      <!--  <work name="hybridSS-48hr" workers="2100" runtime="172800" >   -->
                                              ^^^^
                                              ^^^^
        <work name="hybridSS-48hr" workers="2100" runtime="36000" >
        <operation name="readOP" type="read" ratio="44" config="cprefix=bucket;oprefix=secondary;containers=u(
4,6);objects=u(1,50000000);hashCheck=true" />
            <operation name="listOP" type="list" ratio="15" config="cprefix=bucket;oprefix=secondary;container
s=u(4,6);objects=u(1,50000000);hashCheck=true" />
            <operation name="writeOP" type="write" ratio="36" config="cprefix=bucket;oprefix=secondary;contain
ers=u(1,3);objects=u(1,50000000);sizes=h(1|2|25,2|4|40,4|8|25,8|256|10)KB" />
            <operation name="deleteOP" type="delete" ratio="5" config="cprefix=bucket;oprefix=secondary;contai
ners=u(1,3);objects=u(1,50000000)" />
        </work>
    </workstage>
```

but radosgw is configured to serve only the default 1024 concurrent requests, hence the 503s:

```
ssh <elided>@<elided>.scalelab.redhat.com

# podman ps
CONTAINER ID  IMAGE
   COMMAND               CREATED     STATUS         PORTS       NAMES
...
9a1e75d32bb4  quay.ceph.io/ceph-ci/ceph@sha256:6d5615b2f1b9e54e793b440a9a90c39b014e02f9a47dfe12a90ad4deed4f87e
5  -n client.rgw.rgw...  5 days ago  Up 5 days ago              ceph-ec98cd56-8376-11ec-ae40-000af7995d6c-rgw-
rgws-f22-h05-000-6048r-qvshjs

root@f22-h05-000-6048r:~
# podman exec -it 9a1e75d32bb4 bash
[root@f22-h05-000-6048r /]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 Feb01 ?        00:00:00 /dev/init -- /usr/bin/radosgw -n client.rgw.rgws.f22-h05-0
00-6048r.qvshjs -f --setuser ceph --setgroup ceph --default-log-to-file=false --de
ceph           7       1 27 Feb01 ?        1-14:08:04 /usr/bin/radosgw -n client.rgw.rgws.f22-h05-000-6048r.qv
shjs -f --setuser ceph --setgroup ceph --default-log-to-file=false --default-log-t
root         621       0  0 11:52 pts/0    00:00:00 bash
root         637     621  0 11:52 pts/0    00:00:00 ps -ef
[root@f22-h05-000-6048r /]#

[root@f22-h05-000-6048r /]# find / -name "*.asok"
/run/ceph/ceph-client.rgw.rgws.f22-h05-000-6048r.qvshjs.7.94570918327824.asok
/run/ceph/ceph-osd.93.asok
/run/ceph/ceph-osd.85.asok
...

[root@f22-h05-000-6048r /]# ceph --admin-daemon /run/ceph/ceph-client.rgw.rgws.f22-h05-000-6048r.qvshjs.7.9457
0918327824.asok config show | egrep "front|rgw_thr|concurrent_req"
    "rbd_cache_block_writes_upfront": "false",
    "rgw_frontend_defaults": "beast ssl_certificate=config://rgw/cert/$realm/$zone.crt ssl_private_key=config:
//rgw/cert/$realm/$zone.key",
    "rgw_frontends": "beast endpoint=172.19.43.218:8080",
    "rgw_max_concurrent_requests": "1024",
                                   ^^^^
```

```
                                    ^^^^
    "rgw_thread_pool_size": "512",
```

is it possible to please re-check with either cosbench `workers="1000"` or
rgw `rgw_max_concurrent_requests=3072` & `rgw_thread_pool_size=1536` (and cosbench `workers="2100")
(or `rgw_max_concurrent_requests=2048` & `rgw_thread_pool_size=1024` and cosbench `workers="2000")

#11 - 02/07/2022 05:31 PM - Casey Bodley

thanks Mark! with rgw_max_concurrent_requests=1024, it seems like we should still get consistent performance without this drop-off

is rgw handling this 503 Slow Down case incorrectly? or do you suspect cosbench isn't handling them?

#12 - 02/07/2022 05:37 PM - Matt Benjamin

> I at least would strongly suspect it. I think you'd need to verify by inspection that an RGW cluster in the waning phase is still accessible for other workloads. OTOH, Mark, you've mentioned that memory growth happens on master that we don't see with (properly configured) nautilus. What about quincy?

the memory growth is noticeable only with large objects (24MB in my tests workloads),
the size distribution in this test is only 1MB on 15% of objs : `sizes=h(1|1|50,64|64|15,8192|8192|15,65536|65536|15,1048576|1048576|5)KB`
and excels in the email thread also did not show large RGW memory footprint
hence currently do not suspect that the cause is related to memory.

#13 - 02/08/2022 10:15 AM - Mark Kogan

Casey Bodley wrote:

> thanks Mark! with rgw_max_concurrent_requests=1024, it seems like we should still get consistent performance without this drop-off
>
> is rgw handling this 503 Slow Down case incorrectly? or do you suspect cosbench isn't handling them?

suspect that cosbench is misbehaving when there is a large number of failure.

#14 - 02/08/2022 10:27 AM - Mark Kogan

bringing attention to the balancer (as noted in the email thread)
After discussion with Tim -
The cluster has more that one RGW and the cosbench load should have been distributed between them in which case 2100 req is OK,
but the 503 hint that something is off with the distribution, either cosbench is sending the load to a single RGW instead of a balancer
or that the balancer does not distribute the requests evenly

#15 - 02/10/2022 03:06 PM - Casey Bodley

- Status changed from New to Triaged

- Priority changed from Urgent to High

#16 - 02/10/2022 05:48 PM - Mark Kogan

quick update of the current status (pasting from mail):

```
V. U.
Wed, Feb 9, 9:30 PM (22 hours ago)
to Mark, me, Tim, Casey, Neha, Rachana, Josh

I think it is kind of similar in Cosbench and today morning we verified that Haproxy is configured properly an
d doing balancing fine PFA, Screenshot. Here is how it is configured:
Cosebnch Controller is the guy who starts the workload with given XML and sends these xmls to the driver nodes
 to start the real work.
XML file has localhost:5000 address that is the address for HAPROXY running on each Cosbench driver nodes
Every Cosbench driver node haproxy has all eight RGW configured in a round-robin method.
Cosbench Controller -> Cosbench Driver nodes(haproxy nodes) -> Eight RGW daemons(round-robin)

This looks to be the tuning issue because after tuning the following,
we can see at least for now the 48-hr-hybrid run that was just failing in ~5 hours has crossed ~18 hours.
                                                          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Cosbench workers = 1950
ceph config set global rgw_max_concurrent_requests 2048
ceph config set global rgw_thread_pool_size 1024
```

#17 - 02/11/2022 08:37 PM - Tim Wilkinson

The retesting with the tweaks described above allowed the hybrid-48hr runs to execute for the duration. For the sake of repeatability they are being run again over the weekend. If time allows we may try again with a swift user as opposed to s3 but not committing.

#18 - 02/21/2022 08:35 PM - Tim Wilkinson

Again using rgw_max_concurrent_requests=2048 and rgw_thread_pool_size=1024, the repeatability testing using s3 auth as well, as with swift auth, occurred without issue and the 48hr tests completed.

#19 - 02/21/2022 08:37 PM - Vikhyat Umrao

Tim Wilkinson wrote:

> Again using rgw_max_concurrent_requests=2048 and rgw_thread_pool_size=1024, the repeatability testing using s3 auth as well, as with swift auth, occurred without issue and the 48hr tests completed.

Thank you, Tim.

#20 - 02/21/2022 08:38 PM - Vikhyat Umrao

- Status changed from Triaged to Resolved

Files

w74-48hrHybrid-quincy-smallObjs.png	255 KB	02/03/2022	Tim Wilkinson