Ceph对象存储添加osd导致业务io被长时间block问题

# 问题描述

某项目对象存储对象个数超过1亿3千万，单个bucket接近9千万对象

ceph集群存在1个object omap个数超过5千万，添加osd时会导致业务io长时间block，影响业务正常运行

|  |
| --- |
| ceph health detail  HEALTH\_WARN noout,noscrub,nodeep-scrub flag(s) set; 1 large omap objects  OSDMAP\_FLAGS noout,noscrub,nodeep-scrub flag(s) set  LARGE\_OMAP\_OBJECTS 1 large omap objects  1 large objects found in pool 'clusterB-zone.rgw.buckets.index'  Search the cluster log for 'Large omap object found' for more details. |

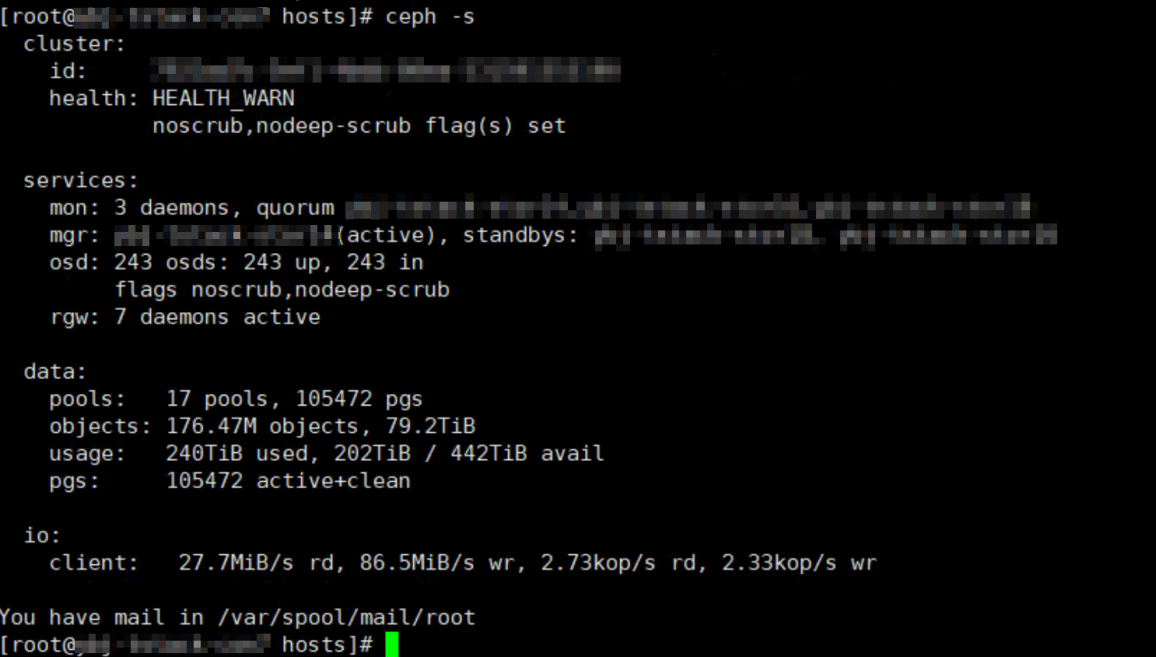
导致这个告警是由于Ceph默认配置

|  |
| --- |
| osd\_deep\_scrub\_large\_omap\_object\_key\_threshold= 2000000 # 200万 |

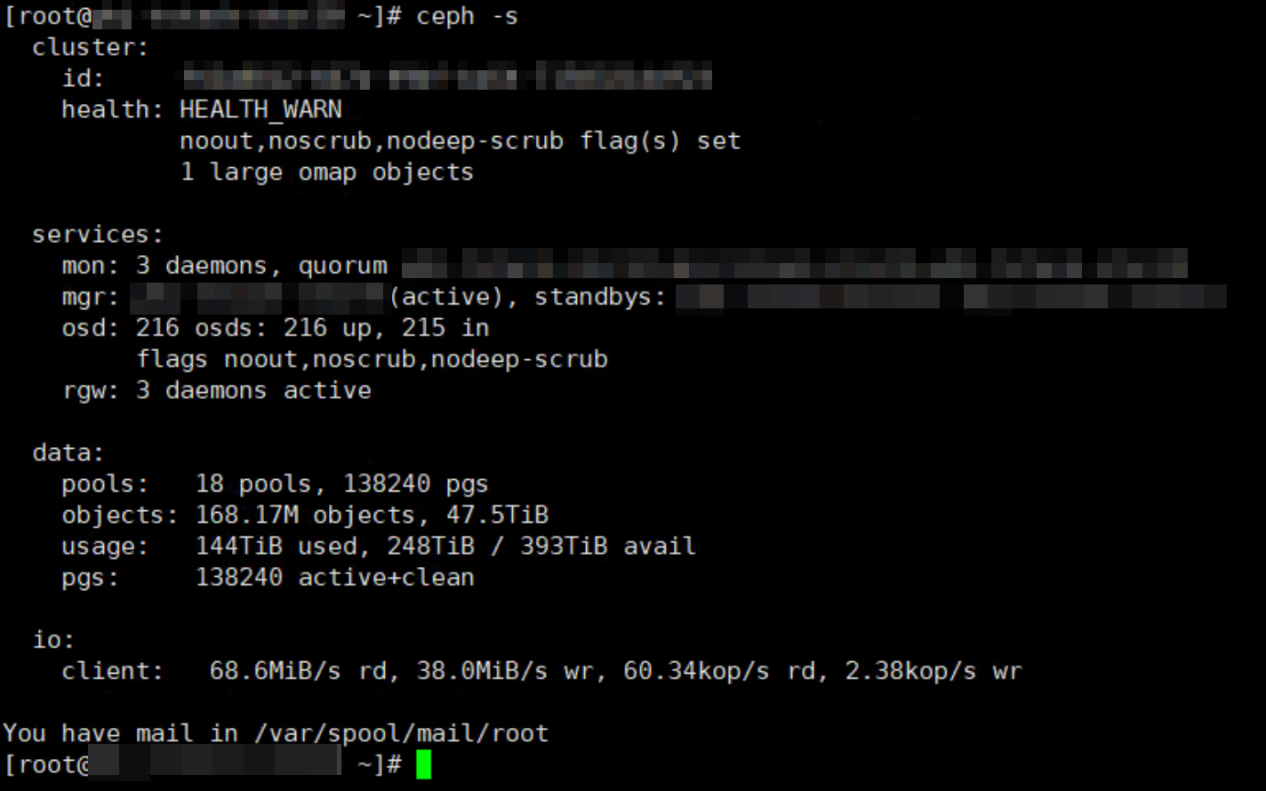
# 环境信息

对象存储multisite，2套集群，A集群正常，B集群有个告警

A集群



B集群



# 问题分析

找到存在large omap object的pgid

准备一个脚本vim large\_obj.py

|  |
| --- |
| import json  import rados  import rbd  ceph\_conf\_path = '/etc/ceph/ceph.conf'  rados\_connect\_timeout = 5  class RADOSClient(object):  def \_\_init\_\_(self,driver,pool=None):  self.driver = driver  self.client, self.ioctx = driver.\_connect\_to\_rados(pool)  def \_\_enter\_\_(self):  return self  def \_\_exit\_\_(self, type\_, value, traceback):  self.driver.\_disconnect\_from\_rados(self.client, self.ioctx)  class RBDDriver(object):  def \_\_init\_\_(self,ceph\_conf\_path,rados\_connect\_timeout,pool=None):  self.ceph\_conf\_path = ceph\_conf\_path  self.rados\_connect\_timeout = rados\_connect\_timeout  self.pool = pool  def \_connect\_to\_rados(self, pool=None):  client = rados.Rados(conffile=self.ceph\_conf\_path)  try:  if self.rados\_connect\_timeout >= 0:  client.connect(timeout=  self.rados\_connect\_timeout)  else:  client.connect()  if self.pool == None:  ioctx = None  else:  ioctx = client.open\_ioctx(self.pool)  return client, ioctx  except rados.Error:  msg = "Error connecting to ceph cluster."  client.shutdown()  raise msg  def \_disconnect\_from\_rados(self, client, ioctx=None):  if ioctx == None:  client.shutdown()  else:  ioctx.close()  client.shutdown()  class cmd\_manager():  def get\_large\_omap\_obj\_poolname(self):  with RADOSClient(RBDDriver(ceph\_conf\_path,rados\_connect\_timeout)) as dr:  result = ''  cmd = '{"prefix": "health", "detail": "detail", "format": "json"}'  result = dr.client.mon\_command(cmd,result)  if result[0] == 0:  res\_ = json.loads(result[1])  if res\_["checks"]['LARGE\_OMAP\_OBJECTS']:  return res\_["checks"]['LARGE\_OMAP\_OBJECTS']['detail'][0]['message'].split("'")[1]  else:  return False  def get\_pg\_list\_by\_pool(self,poolname):  with RADOSClient(RBDDriver(ceph\_conf\_path,rados\_connect\_timeout)) as dr:  result = ''  cmd = '{"prefix": "pg ls-by-pool", "poolstr": "' + poolname + '", "format": "json"}'  result = dr.client.mon\_command(cmd,result)  if result[0] == 0:  return json.loads(result[1])  else:  return False  cmd\_ = cmd\_manager()  poolname = cmd\_.get\_large\_omap\_obj\_poolname()  print "Large omap objects poolname = {0}".format(poolname)  res = cmd\_.get\_pg\_list\_by\_pool(poolname)  for i in res:  if i["stat\_sum"]["num\_large\_omap\_objects"] != 0:  print "pgid={0} OSDs={1} num\_large\_omap\_objects={2}".format(i["pgid"],i["acting"],i["stat\_sum"]["num\_large\_omap\_objects"]) |

查找large omap所在的pg

|  |
| --- |
| python large\_obj.py  Large omap objects poolname = clusterB-x-zone.rgw.buckets.index  pgid=30.91 OSDs=[141, 215, 127] num\_large\_omap\_objects=1 |

查询pg详情

|  |
| --- |
| ceph pg 30.91 query  {  "state": "active+clean",  "peer\_info": [  {  "peer": "215",  "pgid": "30.91",  "stats": {  "last\_deep\_scrub\_stamp": "2020-10-02 02:28:35.978219",  "stat\_sum": {  "num\_bytes": 0,  "num\_objects": 1,  **"num\_large\_omap\_objects": 1**  },  "up": [  141,  215,  127  ],  "acting": [  141,  215,  127  ],  "blocked\_by": [],  "up\_primary": 141,  "acting\_primary": 141  }  }  ]  } |

找到pg所在的3个osd，查看omap文件总大小，osd上的omap过多导致文件很大19G，正常的osd omap文件夹不超过6G

|  |
| --- |
| [root@xxxxx ~]# ssh xxxxx  Warning: Permanently added 'xxxxxxx,x.x.x.x' (ECDSA) to the list of known hosts.  Last login: Thu May 13 15:51:32 2021 from x.x.x.x  [root@xxxxx ~]# du -sh /var/lib/ceph/osd/ceph-215/current/omap/  **16G /var/lib/ceph/osd/ceph-215/current/omap/**  [root@xxxxx ~]# logout  Connection to xxxxx closed.  [root@xxxxx ~]# ssh xxxxx  Warning: Permanently added 'xxxxx,x.x.x.x(ECDSA) to the list of known hosts.  Last login: Thu May 13 16:55:45 2021 from x.x.x.x  [root@xxxxx ~]# du -sh /var/lib/ceph/osd/ceph-127/current/omap/  **18G /var/lib/ceph/osd/ceph-127/current/omap/**  [root@xxxxx ~]# logout  Connection to xxxxx closed.  [root@xxxxx ~]# ssh xxxxx  Warning: Permanently added 'xxxxx,x.x.x.x(ECDSA) to the list of known hosts.  Last login: Fri May 14 18:57:35 2021 from x.x.x.x  [root@xxxxx ~]# du -sh /var/lib/ceph/osd/ceph-141/current/omap/  **19G /var/lib/ceph/osd/ceph-141/current/omap/**  [root@xxxxx ~]# |

查找ceph 所有节点log搜索Large omap object found，找到一个桶分片omap对象过大，超过5千万条数据，占用空间大约11.5G

|  |
| --- |
| 2021-01-26 02:43:53.055519 osd.141 osd.141 x.x.x.x:6802/668087 88512 : cluster [WRN] Large omap object found. Object: 30:8905fecf:::.dir.6210e962-feba-4520-8b2e-6e02ed30540c.131681127.2:head Key count: **53957025** Size (bytes): **12445786687** |

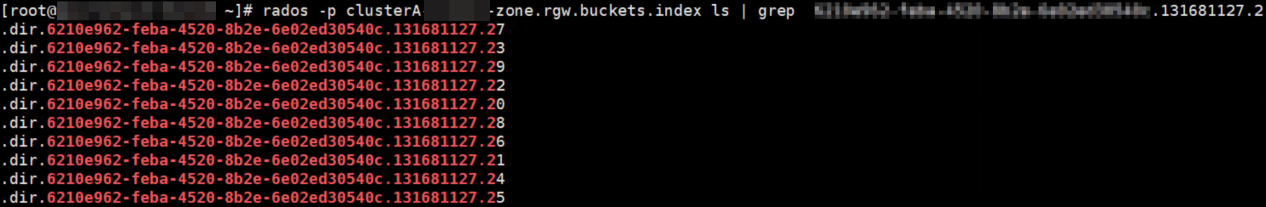
查询可疑对象所在osd

|  |
| --- |
| ceph osd map clusterB-zone.rgw.buckets.index .dir.6210e962-feba-4520-8b2e-6e02ed30540c.131681127.2  osdmap e109640 pool 'clusterB-x-zone.rgw.buckets.index' (30) object '.dir.6210e962-feba-4520-8b2e-6e02ed30540c.131681127.2' -> pg 30.f37fa091 (30.91) -> up ([**141,215,127**], p141) acting ([141,215,127], p141) |

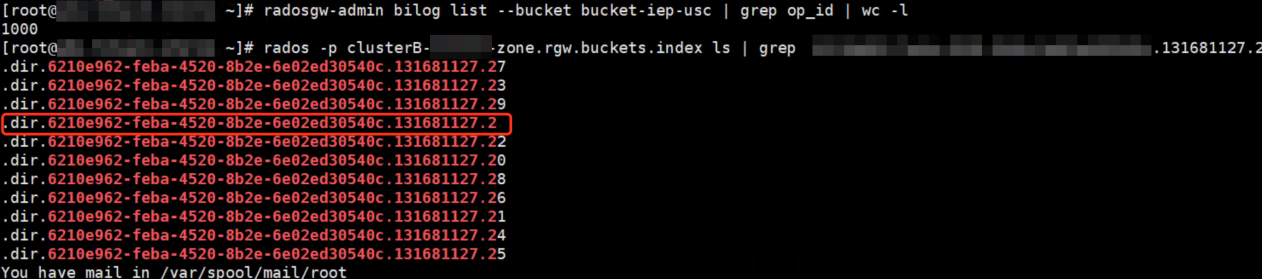
对比2个集群，分析该对象是由于之前执行桶分片操作时残留的索引

.dir.6210e962-feba-4520-8b2e-6e02ed30540c.131681127.2，分别在2个集群上找

A集群查询没有该索引对象



B集群查询索引对象存在，但是桶bucket-iep-usc当前并未使用该索引对象

**解决办法**

删除残留索引对象可临时解决该问题

|  |
| --- |
| rados rm .dir.6210e962-feba-4520-8b2e-6e02ed30540c.131681127.2 -p clusterB-x-zone.rgw.buckets.index |

操作过程中需要注意：

1. 在删除omap数据过程中，由于rocksdb操作时间过长会出现osd down的现象，等删除完osd会自动恢复为up。
2. 删除完再次检查osd omap大小，可以看到明显减小，在合理范围内；
3. 删完数据集群的告警one large omap objects并不会自动消失，手动执行deep-scrub即可，ceph pg deep-scrub 30.91
4. 另外在实施过程中遇到备份数据耗时过长问题，通过命令行大约每秒只能备份100条数据，因此不建议通过命令行备份

|  |
| --- |
| mkdir objomap.bak  cd objomap.bak  poolname="clusterB-x-zone.rgw.buckets.index"  objname=".dir.6210e962-feba-4520-8b2e-6e02ed30540c.131681127.2"  rados -p $poolname listomapkeys ${objname} > ${objname}.key.omap  for id in `cat ${objname}.key.omap`;do rados -p $poolname getomapval ${objname} $id $id;done |

使用脚本备份，测试110万条数据1分钟完成，5千万条数据预估45分钟备份完成，参考脚本

|  |
| --- |
| #!/usr/bin/env python  # -\*- coding: UTF-8 -\*-  import os  from rados import (Rados, ReadOpCtx, ObjectNotFound)  MAX\_SIZE = 100000  def backup\_obj\_omap(pool\_name, object\_name):  cluster = Rados(conffile='/etc/ceph/ceph.conf')  cluster.connect()  ioctx = cluster.open\_ioctx(pool\_name)  start\_str = ""  count = 0  with ReadOpCtx(ioctx) as read\_op\_inst:  while True:  iter, \_ = ioctx.get\_omap\_vals(read\_op\_inst, start\_str, "", MAX\_SIZE)  try:  ioctx.operate\_read\_op(read\_op\_inst, object\_name)  except ObjectNotFound:  return  data = list(iter)  if len(data) == 0:  break  dir\_name = str(count)  os.makedirs(dir\_name)  for k, v in data:  # print(k, v)  f = open(os.path.join(str(count), k), "w")  f.write(v)  f.close()  start\_str = k  count += 1  if ioctx is None:  cluster.shutdown()  else:  ioctx.close()  cluster.shutdown()  if \_\_name\_\_ == '\_\_main\_\_':  backup\_obj\_omap('images', 'test') |

恢复脚本:参考备份脚本修改

# 建议

1.修改告警阈值osd\_deep\_scrub\_large\_omap\_object\_key\_threshold=200000，避免当omap数据过多时导致业务不可用

由于缺省告警阈值单个object 2million k/v太高，当具有大量omap数据的存储桶索引执行recovery时会阻塞客户端访问该桶内数据，最新代码已降低该值到20万，详见:

https://tracker.ceph.com/issues/40583

该patch已合入到L版v12.2.13

https://github.com/ceph/ceph/pull/29175

2.修改桶配额，限制用户一直往同一个桶内上传过多对象，从业务层进行优化，对象过多的情况考虑分开存放到多个桶内。

# 参考

<https://access.redhat.com/solutions/5703641>

<https://cloud.tencent.com/developer/article/1400660>