4. 第四章 glusterfs 的特性

从这一章开始，主要分享一些 glusterfs 的特性，例如限制 volume 容量的 quota,快照 snapshot 和 add-brick 的 rebalance 等 ,这些功能都有着不同的特点与应用场景，下面将进行详细地讲解一下。

4.1. quota 容量限制

4.1.1. 开启 quota 功能

Glusterfs 中的 quota 功能其实就是在对 volume 进行容量限制 ,使用起来也比较简单，下面来简单尝试一下。

```
[root@gfs03 ~]# gluster volume quota  test-quota enable
volume quota : success

[root@gfs03 ~]# gluster volume quota  test-quota limit-usage / 1GB
volume quota : success

[root@gfs03 ~]# gluster volume quota  test-quota list /
  Path    Hard-limit  Soft-limit     Used   Available  Soft-limit exceeded? Hard-limit exceeded?
  -----------------------------------------------------------------------------
/          1.0GB     80%(819.2MB)   0Bytes  1.0GB              No                   No

[root@gfs03 ~]# gluster volume info test-quota

Volume Name: test-quota
Type: Replicate
Volume ID: cfc66790-7e2f-45e1-b188-4f6a9a3bb210
Status: Started
Snapshot Count: 0
```

```
Number of Bricks: 1 x 3 = 3
Transport-type: tcp
Bricks:
Brick1: 192.168.0.110:/glusterfs/test-quota
Brick2: 192.168.0.111:/glusterfs/test-quota
Brick3: 192.168.0.112:/glusterfs/test-quota
Options Reconfigured:
features.quota-deem-statfs: on
features.inode-quota: on
features.quota: on
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

这里开启了容量限制功能，并且对 volume 的根目录大小限制为 1GB，那么这里查看 volume info 信息的时候 ,可以留意到多了两个特性 ,就是和 quota 有关的。

那么下面尝试写入数据进行测试。

```
[root@gfsclient01 ~]# mount -t glusterfs 192.168.0.110:test-quota /mnt/test-quota

[root@gfsclient01 ~]# cp -rp glusterfs-9.2.zip /mnt/test-quota/

[root@gfsclient01 ~]# du -sh /mnt/test-quota/
4.9M    /mnt/test-quota/

[root@gfs03 ~]# gluster volume quota  test-quota list /
 Path      Hard-limit  Soft-limit    Used  Available  Soft-limit exceeded? Hard-limit exceeded?
-------------------------------------------------------------------------------
/             1.0GB     80%(819.2MB)    4.8MB 1019.2MB             No                   No
```

这里导入了一个比较小的文件，然后再次使用 list 的时候发现使用量已经改

变了。

4.1.2. quota 对不同 volume 的使用

那么这里前面测试了一下开启 quota 功能的使用，那么这里对于复制卷，到底 quota 的容量是总容量呢？还是每个 brick 的容量？而对于冗余卷来说，因为这里会对数据进行拆分，又是怎样的呢？下面来做一个简单的测试对比。

```
root@gfs01:~# gluster volume info test-replica

Volume Name: test-replica
Type: Replicate
Volume ID: d2614e89-9aba-46f6-bf04-984782ac6d6f
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 3 = 3
Transport-type: tcp
Bricks:
Brick1: 10.0.12.2:/glusterfs/test-replica
Brick2: 10.0.12.9:/glusterfs/test-replica
Brick3: 10.0.12.12:/glusterfs/test-replica
Options Reconfigured:
features.quota-deem-statfs: on
features.inode-quota: on
features.quota: on
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

这里有一个简单的 test-replica 的 3 副本的 volume ,然后下面开启 quota 并且限制容量为 2GB,并且挂载之后导入一个镜像查看一下效果。

```
root@gfs01:~# gluster volume quota test-replica list /
Path     Hard-limit  Soft-limit   Used  Available  Soft-limit exceeded? Hard-limit exceeded?
```

```
3.  ---------------------------------------------------------
    ---------------------
4.  /            2.0GB      80%(1.6GB)    0Bytes    2.0GB
         No                      No
5.
6.  root@gfs01:~# ls /mnt/test-replica/
7.
8.  root@gfs01:~# cp CentOS-8.3.2011-x86_64-minimal.iso   /mnt
    /test-replica/
9.
10. root@gfs01:~# du -sh /glusterfs/test-replica
11. 1.8G    /glusterfs/test-replica
12.
13. root@gfs01:~# du -sh /mnt/test-replica/
14. 1.8G    /mnt/test-replica/
15.
16. root@gfs01:~# gluster volume quota test-replica list /
17.  Path      Hard-limit  Soft-limit    Used  Available  Sof
    t-limit exceeded? Hard-limit exceeded?
18. ---------------------------------------------------------
    ----------------------
19. /              2.0GB      80%(1.6GB)     1.7GB 268.0MB
          Yes                      No
```

这里可以看到 brick 中的容量大小是和 quota 大小一样的 ,说明这里 quota 对复制卷的限制的容量大小就是每一个 brick 的大小，下面再看一下其他两个 brick 其实也是一样的。那么这里对于 disperse 冗余卷呢？下面也简单测试一下。

```
1.  root@gfs01:~# gluster volume info test-disperse
2.
3.  Volume Name: test-disperse
4.  Type: Disperse
5.  Volume ID: 529c1b28-d49b-4058-a8ef-cc43d265061a
6.  Status: Started
7.  Snapshot Count: 0
8.  Number of Bricks: 1 x (2 + 1) = 3
9.  Transport-type: tcp
10. Bricks:
11. Brick1: 10.0.12.2:/glusterfs/test-disperse
```

```
Brick2: 10.0.12.9:/glusterfs/test-disperse
Brick3: 10.0.12.12:/glusterfs/test-disperse
Options Reconfigured:
features.quota-deem-statfs: on
features.inode-quota: on
features.quota: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on

root@gfs01:~# gluster volume quota test-disperse list /
Path      Hard-limit  Soft-limit    Used  Available  Soft-limit exceeded? Hard-limit exceeded?
-----------------------------------------------------------------------------
/             N/A       N/A        N/A     N/A        N/A                  N/A

root@gfs01:~# cp CentOS-8.3.2011-x86_64-minimal.iso  /mnt/test-disperse/

root@gfs01:~# gluster volume quota test-disperse list /
Path         Hard-limit  Soft-limit      Used  Available  Soft-limit exceeded? Hard-limit exceeded?
-----------------------------------------------------------------------------
/               2.0GB     80%(1.6GB)    1.7GB 268.0MB             Yes                  No

root@gfs01:~# du -sh /mnt/test-disperse/
1.8G    /mnt/test-disperse/

root@gfs01:~# du -sh /glusterfs/test-disperse/
892M    /glusterfs/test-disperse/
```

从这里可以看到，disperse 冗余卷的数据分布是不一样的，这里会对文件进行拆分，那么这里的 quota 大小，其实就是数据的完整大小的，而不是单个 brick 的大小了 ,这里就要注意区别了 , 下面再看看其他两个 brick 的大小情况。

```
root@gfs02:~# du -sh /glusterfs/test-disperse/
892M    /glusterfs/test-disperse/
```

```
3.
4.  root@gfs03:~# du -sh /glusterfs/test-disperse/
5.  892M    /glusterfs/test-disperse/
```

所以这里对于不同类型的 volume 使用 quota 大小是有区别的，那么这里对于分布式复制卷呢？会不会又不一样，大家可以自行测试一下。

4.1.3. quota 真的能限制大小吗？

那么下面尝试一下将一个超过容量限制的文件存放进 volume 里面 ,看看会有怎样的效果。

```
1.  [root@gfsclient01 ~]# du -sh ubuntu-18.04.3-desktop-amd64.
    iso
2.  2.0G    ubuntu-18.04.3-desktop-amd64.iso
3.
4.  [root@gfsclient01 ~]# cp ubuntu-18.04.3-desktop-amd64.iso
     /mnt/test-quota/
5.  cp: error writing ‘/mnt/test-quota/ubuntu-18.04.3-desktop
    -amd64.iso’: Disk quota exceeded
6.  cp: failed to extend ‘/mnt/test-quota/ubuntu-18.04.3-desk
    top-amd64.iso’: Disk quota exceeded
7.  cp: failed to close ‘/mnt/test-quota/ubuntu-18.04.3-deskt
    op-amd64.iso’: Disk quota exceeded
8.
9.
10. [root@gfs03 ~]# gluster volume quota  test-quota list /
11. Path       Hard-limit  Soft-limit      Used  Available  S
    oft-limit exceeded? Hard-limit exceeded?
12. ---------------------------------------------------------
    -------------------------------
13. /            1.0GB     80%(819.2MB)    1.1GB  0Bytes
           Yes                   Yes
```

这里在客户端放入一个 2G 大小的镜像文件，而 volume 容量才 1GB，并且已经使用一些空间了 ,因此这里放入的文件最终是失败的 ,会提示 Disk quota

exceeded，也就是超过容量限制了，而这时候再次查看 volume 的容量使用，会发现已经超过了容量限制了。因此在使用 quota 的时候，要对这一点比较注意，并不是使用 quota 就一定可以绝对限制容量大小的。

那么这时候再来看 brick 的日志，会发现很多报错了。

```
[root@gfs01 bricks]# tail -n 10 -f glusterfs-test-quota.log
........
[2021-06-15 14:54:03.263585 +0000] W [socket.c:767:__socket_rwv] 0-tcp.test-quota-server: readv on 192.168.0.112:49114 failed (No data available)
[2021-06-15 14:54:03.263628 +0000] I [MSGID: 115036] [server.c:500:server_rpc_notify] 0-test-quota-server: disconnecting connection [{client-uid=CTX_ID:b15f6358-7988-40f9-bf47-13f00016774a-GRAPH_ID:0-PID:2005-HOST:gfs03-PC_NAME:test-quota-client-0-RECON_NO:-0}]
[2021-06-15 14:54:03.263740 +0000] I [MSGID: 101055] [client_t.c:397:gf_client_unref] 0-test-quota-server: Shutting down connection CTX_ID:b15f6358-7988-40f9-bf47-13f000167774a-GRAPH_ID:0-PID:2005-HOST:gfs03-PC_NAME:test-quota-client-0-RECON_NO:-0
[2021-06-15 14:54:04.129323 +0000] E [MSGID: 115067] [server-rpc-fops_v2.c:1324:server4_writev_cbk] 0-test-quota-server: WRITE info [{frame=13323}, {WRITEV_fd_no=0}, {uuid_utoa=26d275a0-510b-47c5-aa8b-c93fcbd8f1a8}, {client=CTX_ID:8616da2c-1d9b-4952-809d-ecdafda5fe7d-GRAPH_ID:0-PID:1616-HOST:gfsclient01-PC_NAME:test-quota-client-0-RECON_NO:-0}, {error-xlator=test-quota-quota}, {errno=122}, {error=Disk quota exceeded}]
..........
```

4.1.4. quota 监控

那么这里如果想要监控容量的话，操作系统中如果分区的容量不是独占的，那么是无法正常监控的，如下所示。

```
[root@gfs03 ~]# df -h /glusterfs/test-quota
```

```
Filesystem                Size  Used Avail Use% Mounted on

/dev/mapper/centos-root    50G  3.8G   47G   8% /

[root@gfs03 ~]# gluster volume get test-quota quota-deem-statfs
Option                                  Value
------                                  -----
features.quota-deem-statfs              on
```

这里因为系统中的使用 df -h 的结果是不准确的，因此这里可以考虑使用一些脚本的方式进行监控。

这里要实现的步骤如下所示：

1. 遍历所有的 volume,然后判断是否开启 quota 功能，如果开启则跳转 2 ，否则下一个 volume，遍历完成所有 volume 之后，进入步骤 5。

2. 这里还要进一步检查，quota list 中的数据是否为 Null，如果是则进行下一个 volume 遍历，回到步骤 1，否则进入下一个步骤

3. 获取 quota list 中的信息，这里获取 Hard-limit 和 Used 下的大小，然后进行数值单位归一化处理，也就是防止 TB 与 MB 的单位的数值进行直接比较，然后进入下一步。

4. 判断计算使用率是否超过阈值，如果超过则累计告警信息，否则回到步骤 1 。

5. 遍历结束后，判断告警信息中是否为空，如果不为空则证明有 volume 要超过阈值要进行告警，输出调用告警函数。

这里的步骤比较简单，不过主要是一些细节可能需要注意一下，例如过滤获取 quota 信息中的单位，要对单位进行归一化处理。下面给出一个简单的监控

告警脚本。

```
#!/bin/bash

volAterStr=""
#这两个变量用于检查数据容量是否有不一致要进行转换的
volQuotaTotalRatio=1
volQuotaUsedRatio=1
#这个变量用于获取当前 vol 的 quota 信息
volQuotaInfo=""
#告警阈值
volLimitAlter=0.7



#这个函数用于对 quota 中的单位显示进行归一化
#方便进行统一比较数值大小.
function volSizeChange(){
  unix=$1
  volRatio=1

  if [  $unix == "GB"  ]
  then
     volRatio=1024
  elif [ $unix == "TB" ]
  then
     volRatio=1048576
  elif [ $unix == "MB" ]
  then
     volRatio=1
  elif [ $unix == "KB"  ]
  then
     volRatio=0.001
  fi

  param=$2
  if [ $param == "total"  ]
  then
     volQuotaTotalRatio=$volRatio
```

```
38.   else
39.      volQuotaUsedRatio=$volRatio
40.   fi
41.
42. }
43.
44.
45. #这个函数用于获取 quota 信息的
46. #有可能出现 quota command failed 报错
47. #入参是 volume 名称
48. volGetQuota(){
49.   volQuotaInfo=" "
50.   num=1
51.   volume=$1
52.
53.   while :
54.   do
55.       volQuotaInfo=`sudo gluster volume quota $volume list
    /`
56.       checkInfo=`sudo echo -e "$volQuotaInfo\n" |grep "Har
   d-limit"`
57.
58.
59.      if [ $num -gt 10  ]
60.      then
61.                    quotaError=`sudo  echo  -e  "\n  当  前
   volume: $volume  尝试获取 quota 超过次数失败!!!!\n"`
62.          volAlterStr=${volAlterStr}${quotaError}
63.          return 1
64.      elif [ -z "$checkInfo"  ]
65.      then
66.              volQuotaInfo=""
67.          sudo echo -e "\n $volume 尝试 $num 次获取 quota 信
   息...."
68.      else
69.          return 0
70.      fi
```

```
71.
72.         num=`expr $num + 1`
73.
74.     done
75.
76. }
77.
78.
79. #获得 vol 的 quota 容量信息进行比较的.
80. #如果超过比例的话，那么这里将会累计告警信息.
81. #待全部 vol 遍历完成再统一发送告警,避免过于频繁发送.
82. volQuota(){
83.
84.         volume=$1
85.
86.         volGetQuota $volume
87.         quota=$volQuotaInfo
88.
89.         #这里要对计算单位进行归一化处理
90.         volQuotaTotalRatio=1
91.         quotaTotal=`sudo echo -e "$quota\n"  | grep "80%" |
    awk -F ' ' '{print $2}' | tr -cd "[0-9.]"`
92.         quotaTotalUnix=`sudo echo -e "$quota\n"  | grep "80
    %" |awk -F ' ' '{print $2}' | tr -d "0-9."`
93.         volSizeChange $quotaTotalUnix "total" $volume
94.         quotaTotalNew=`awk -v x=$quotaTotal -v y=$volQuotaT
    otalRatio 'BEGIN{printf "%.2f\n",x*y}'`
95.
96.
97.         volQuotaUsedRatio=1
98.         quotaUsed=`sudo echo -e "$quota\n"  | grep "80%"  |
    awk -F ' ' '{print $4}' | tr -cd "[0-9.]"`
99.         quotaUsedUnix=`sudo echo -e "$quota\n"  | grep "80%
    "  |awk -F ' ' '{print $4}' | tr -d "0-9."`
100.        volSizeChange $quotaUsedUnix  "used" $volume
101.        quotaUsedNew=`awk -v x=$quotaUsed -v y=$volQuotaUse
    dRatio 'BEGIN{printf "%.2f\n",x*y}'`
102.
103.        #判断是否为空或者 null
```

```
104.      if [ -z "$quotaTotal" ]
105.      then
106.          sudo echo -e "$1 volume 已经开启 quota 功能，但是总
   容量为 Null "
107.      elif [ -z "$quotaUsed" ]
108.      then
109.          sudo echo -e "$1 volume 已经开启 quota 功能，但是使
   用量为 Null "
110.      else
111.          percent=`awk 'BEGIN{printf "%.2f\n",'$quotaUsed
   New'/'$quotaTotalNew'}'`
112.          #如果 num1>num2,则为 1,否则为 0
113.          result=`awk -v num1=$volLimitAlter  -v num2=$pe
   rcent 'BEGIN{print(num1>num2)?"1":"0"}'`
114.          if [ $result  -eq 0 ]
115.          then
116.             #告警信息累计
117.              quotaTotalWithSize=`sudo echo -e "$quota\n"
    | grep "80%" |awk -F ' ' '{print $2}' `
118.              quotaUsedWithSize=`sudo echo -e "$quota\n"
   | grep "80%" |awk -F ' ' '{print $4}'`
119.              volumeAlterStr="\n\nvolume: $1\n  volQuotaTo
   tal:  $quotaTotalWithSize\n  volQuotaUsedWithSize: $quotaU
   sedWithSize\n  volUsedPercent: $percent\n"
120.             #这里拼接告警信息
121.              volAlterStr=${volAlterStr}${volumeAlterStr}
122.          fi
123.      fi
124.
125.}
126.
127.
128.#遍历 vol，获取每个 vol 的 quota 信息
129.volQuery(){
130.   volList=`sudo gluster volume list`
```

```
    for vol in $volList
    do
        #检查 vol quota 功能是否开启
     #这里可以通过检查 vol 的 info 信息中是否会 quota 相关特性,并且该特性是 on 的
        features_quota=`sudo gluster volume info $vol  |grep "features.quota" | awk -F ' ' '{print $NF}' |grep "on"`
        if [ -z "$features_quota"  ]
     then
                    sudo echo -e "\n$vol quota disable"
            continue
     else
            #开启 quota 的 vol 将进行检查容量使用情况
            volQuota "$vol"
        fi
    done


    #这里遍历完所有的 volume
    #如果累计的告警信息不为空，那么证明有 vol 要进行告警
    if [ -n "$volAlterStr" ]
    then
        sudo echo -e "\n  alter info: $volAlterStr  \n"
    fi
}


volQuery
```

这里脚本执行的结果如下所示。

```
[root@gfs03 ~]# bash gfs-quota.sh

test-arbiter quota disable

```

```
5.  test-disperse quota disable
6.
7.  test-replica quota disable
8.
9.    alter info:
10.
11. volume: test-quota
12. VolQuotaTotal: 1.0GB
13.  volQuotaUsedWithSize: 1.1GB
14.  volUsedPercent: 1.10
```

对于 quota 的监控，因为如果 volume 不是使用 lvm2 创建的话，那么文件系统中显示的容量大小就是操作系统全部的容量集合 ,因此没有办法准确地显示容量限制大小，所以可以考虑使用命令获取的方式进行监控。

当然这里还有一种操作方式，就是操作系统层面对分区做好容量限制范围，那样每个 volume 的 brick 对应一个分区，但是这种操作相对麻烦一点，可以根据业务场景来进行操作。