4.6. 麻烦的扩缩容

在 glusterfs 当中，如果 glusterfs 的 volume 是使用 heketi 创建的，那么就无法正常地使用 quota 进行容量限制了，因为底层使用 lvm2 的时候，如果要扩缩容，有两种办法，一种是对底层的 vg 和 lv 进行处理，但是这种方法风险比较高，一旦操作不慎，会直接影响原来的数据，而且底层的 vg 和 lv 容量改变，信息还需要同步到上层的应用中，会比较麻烦；第二种方法，就是使用 glusterfs 的 add-brick 功能，也就是增加 brick 的方式来进行扩缩容处理，下面来了解一下。

4.6.1. add-brick 操作

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

Volume Name: test-event
Type: Distribute
Volume ID: 4ff63ab9-561a-4c32-a4b9-60c5bda0c6e9
Status: Created
Snapshot Count: 0
Number of Bricks: 1
Transport-type: tcp
Bricks:
Brick1: 10.0.12.9:/glusterfs/test-event
Options Reconfigured:
nfs.disable: on
transport.address-family: inet
storage.fips-mode-rchecksum: on


root@gfs01:~# gluster volume add-brick test-event 10.0.12.2:/glusterfs/test-event force
volume add-brick: success

root@gfs01:~# gluster volume info test-event

```

```
Volume Name: test-event
Type: Distribute
Volume ID: 4ff63ab9-561a-4c32-a4b9-60c5bda0c6e9
Status: Created
Snapshot Count: 0
Number of Bricks: 2
Transport-type: tcp
Bricks:
Brick1: 10.0.12.9:/glusterfs/test-event
Brick2: 10.0.12.2:/glusterfs/test-event
Options Reconfigured:
nfs.disable: on
transport.address-family: inet
storage.fips-mode-rchecksum: on
```

这里有一个单 brick 的卷 test-event,然后使用了 add-brick 进行扩容，那么这里扩容之后要 注意，数据是否要进行 rebalance，因为对于数据的分布如果不进行重平衡 ,那么可能后续会造成数据倾斜 ,也就是部分 brick 的数据很多，但是新添加的 brick 可能数据很少的现象，而关于 rebalance 的问题，后面还会继续了解。

4.6.2. remove-brick 操作

有了扩容的操作，那么一般自然也会有减少 brick 的，而这个操作可能在实际的生产环境中会使用频率比较少 ,但是这里也有一些值得注意的问题，下面来了解一下。

```
root@gfs01:~# mount -t glusterfs 10.0.12.9:test-event /mnt/test-event

root@gfs01:~# ls /mnt/test-event/

root@gfs01:~# cd /mnt/test-event/

```

```
root@gfs01:~# cat  /mnt/test-event/smallFile.sh
#!/bin/bash

for((i=1;i<=10000;i++));
do
    touch $i.txt
    date > $i.txt
    #echo $(expr $i \* 3 + 1);
done

root@gfs01:~# ls /mnt/test-event/ |wc -l
10001
```

为了测试缩容的效果，这里弄了一万个小文件，然后再执行 remove-brick，在执行之前，这里再核实一下不同的 brick 的文件数量分布情况。

```
root@gfs03:~# ls /glusterfs/test-event/ |wc -l
4989

root@gfs03:~# hostname -i
10.0.12.9

root@gfs02:~# hostname -i
10.0.12.2

root@gfs02:~# ls /glusterfs/test-event/ |wc -l
5012
```

这里可以看到，在该 volume 中，文件不是和复制卷一样的，这里也没有绝对平均地分配文件数量，那么下面进行 remove-brick 操作尝试。

```
root@gfs01:/mnt/test-event# gluster volume remove-brick test-event 10.0.12.9:/glusterfs/test-event force
Remove-brick force will not migrate files from the removed bricks, so they will no longer be available on the volume.
Do you want to continue? (y/n) y
volume remove-brick commit force: success

```

```
root@gfs01:/mnt/test-event# gluster volume info test-event

Volume Name: test-event
Type: Distribute
Volume ID: 4ff63ab9-561a-4c32-a4b9-60c5bda0c6e9
Status: Started
Snapshot Count: 0
Number of Bricks: 1
Transport-type: tcp
Bricks:
Brick1: 10.0.12.2:/glusterfs/test-event
Options Reconfigured:
performance.client-io-threads: on
nfs.disable: on
transport.address-family: inet
storage.fips-mode-rchecksum: on

root@gfs01:/mnt/test-event# gluster volume rebalance  test-event status
volume rebalance: test-event: failed: Volume test-event is not a distribute volume or contains only 1 brick.
Not performing rebalance
```

这里可以看到，因为只有一个 brick，那么是无法进行 rebalance 的，同时这里如果只剩下了最后一个 brick，会有数据丢失的风险的，因此要特别注意。

4.6.3. replace-brick 操作

这里的除了增加和删除之外，其实还有一个叫 replace-brick 的操作，这里在什么时候需要使用该操作呢？主要是在当前节点有问题的时候，想转移这个 brick 的数据到另外一个节点上，那么下面可以了解一下。

```
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


root@gfs01:~# gluster volume replace-brick test-replica 10.0.12.2:/glusterfs/test-replica  10.0.12.2:/glusterfs/test-replica-new commit force
volume replace-brick: success: replace-brick commit force operation successful


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-new
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
```

那么注意，这里因为使用 replace-brick 之后，因为不是分布式卷，因此是无法进行 rebalance 的，下面可以看到内容。

```
root@gfs01:~# gluster volume rebalance test-replica status
volume rebalance: test-replica: failed: Volume test-replica is not a distribute volume or contains only 1 brick.
Not performing rebalance
```

那么这里还可以留意一下是否数据已经转移了。

```
root@gfs02:~# ls -l /glusterfs/test-replica
total 1822728
-rw-r--r-- 2 root root 1866465280 Jun 23 20:51 CentOS-8.3.2011-x86_64-minimal.iso

root@gfs02:~# ls -l /glusterfs/test-replica-new/
total 1822728
-rw-r--r-- 2 root root 1866465280 Jun 23 20:51 CentOS-8.3.2011-x86_64-minimal.iso

root@gfs02:~# md5sum /glusterfs/test-replica/CentOS-8.3.2011-x86_64-minimal.iso
8934d42a86d8589342ac9bdfec82d6b4  /glusterfs/test-replica/CentOS-8.3.2011-x86_64-minimal.iso

root@gfs02:~# md5sum /glusterfs/test-replica-new/CentOS-8.3.2011-x86_64-minimal.iso
8934d42a86d8589342ac9bdfec82d6b4  /glusterfs/test-replica-new/CentOS-8.3.2011-x86_64-minimal.iso

root@gfs02:~# hostname -i
10.0.12.2
```

4.6.4. rebalance 很重要

那么提到了扩缩容的问题，就不得不说一下 rebalance 重平衡了，所谓的重平衡，这里就是对数据的分布进行重新的均衡，这样的话可以保证数据不会出现明显的数据倾斜情况，下面可以进行测试一下。

```
root@gfs01:~# gluster volume create rebalance-test replica 3 10.0.12.{2,9,12}:/glusterfs/rebalance-01 force
volume create: rebalance-test: success: please start the volume to access data

root@gfs01:~# gluster volume start rebalance-test
volume start: rebalance-test: success

root@gfs01:~# mkdir -p /mnt/rebalance-test

root@gfs01:~# mount -t glusterfs 10.0.12.2:rebalance-test /mnt/rebalance-test

root@gfs01:~# cp  /mnt/test-rebalance/test.sh  /mnt/rebalance-test/

root@gfs01:~# cd /mnt/rebalance-test/

root@gfs01:/mnt/rebalance-test# ls
test.sh

root@gfs01:/mnt/rebalance-test# cat test.sh
#!/bin/bash
for ((i=1; i<=10000; i++))
do
  sudo touch $i.txt
  sudo date > $i.txt
done

root@gfs01:/mnt/rebalance-test# bash test.sh

root@gfs01:/mnt/rebalance-test#    ls -l |wc -l
10002
```

这里首先创建一个 3 副本的复制卷，然后使用脚本创建了 1 万个小文件，接着下面就是使用 add-brick 进行增加 brick，然后进行 rebalance 操作。

```
root@gfs01:/mnt/rebalance-test# gluster volume add-brick rebalance-test 10.0.12.{2,9,12}:/glusterfs/rebalance-02  force
volume add-brick: success

root@gfs01:/mnt/rebalance-test# gluster volume rebalance rebalance-test start
volume rebalance: rebalance-test: success: Rebalance on rebalance-test has been started successfully. Use rebalance status command to check status of the rebalance process.
ID: 39101691-e7d7-4e51-847a-4fc2845eea3a

root@gfs01:/mnt/rebalance-test# gluster volume rebalance rebalance-test status
    Node Rebalanced-files size  scanned failures skipped  status  run time in h:m:s
    ---------      -----------   -----------   -----------
   10.0.12.9  21 672Bytes   506    0    0   in progress        0:00:04
    gfs02   22   704Bytes    504   0    0  in progress       0:00:04
  localhost 8   576Bytes 503    0    0   in progress        0:00:04
The estimated time for rebalance to complete will be unavailable for the first 10 minutes.
volume rebalance: rebalance-test: success
```

从这里可以看到,重平衡已经启动了，那么这里数据最后的分布会变成怎样呢？

```
root@gfs02:~# ls -l /glusterfs/rebalance-01/ | wc -l
4990
root@gfs02:~# ls -l /glusterfs/rebalance-02/ | wc -l
5013
```

另外对于 volume 的 rebalance 的操作，这里是有两种情况的，一种是像上面的把数据全部重新平衡，包括元数据，还有一种是单纯平衡元数据的，这里

可以根据业务场景的不同进行选择。

对于 rebalance 这里，还有一点需要注意的是，force 选项并不是强制的，加上 force 的话会考虑 brick 的数据分布情况，也就是说，如果不加上 force 的话,那么 add-brick 之后有可能数据并不会一定进行迁移和平衡的，这一点可以查看官方 issue 编号为 2571 的内容，本人在做测试的时候曾经遇到过。

何翔宇