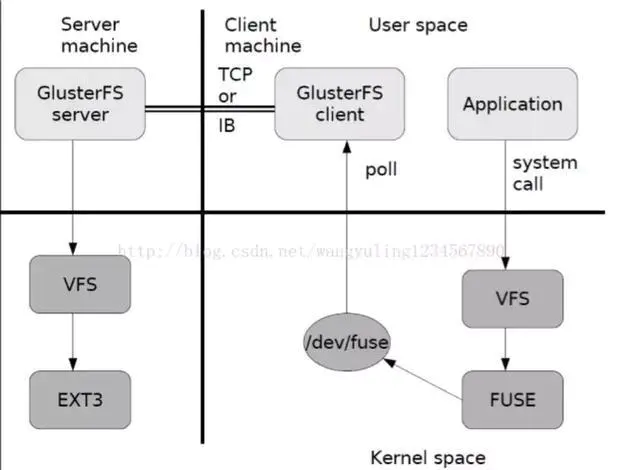
#### 一、Glusterfs集群的五种模式



附件1

如果创建的brick在系统盘，在gluster默认情况下是不允许的，如果需要使用在创建卷时在后面加上force

##### 1.分布卷

文件分布在不同的块服务器（文件1可分布在服务器1或2，但不能两台同时存在，没有冗余），将完整的数据哈希成两份，分布存在于两个brick，以文件为单位的哈希分配

更容易和廉价的扩展卷的大小

单独故障会造成数据丢失

eg:

gluster volume create new-volume server1:/data/brick1/exp2 server2:/data/brick1/exp2

[root@lvs.17.sg1.test disperse]# pwd

/data0/disperse

[root@lvs.17.sg1.test disperse]# touch testfile{1,2,3,4,5,6,7,8,9,10}

[root@lvs.17.sg1.test disperse]# ls

testfile1  testfile10  testfile2  testfile3  testfile4  testfile5  testfile6          testfile7  testfile8  testfile9

我们去192.168.21.18和19的对应的brick上看分别存储的数据。

19：

[root@lvs ~]# ls /data0/gluster/data1/

testfile1  testfile10  testfile2  testfile3  testfile4  testfile6  testfile8

18：

[root@lvs ~]# ls /data0/gluster/data1/

testfile5  testfile7  testfile9

##### 2.复制卷

类似于raid1，可以进行三复制或两复制，所有的块服务器都保持一个精确的副本，即使一个块损坏，仍可以从它的复制块访问数据，具有更高的可靠性和冗余性

eg:

gluster volume create test-volume3 replica 2 transport tcp server1:/data/brick1/test1/ server2:/data/brick1/test1/

gluster volume create test-volume3 replica 3 transport tcp server1:/data/brick1/test1/ server2:/data/brick1/test1/ server3:/data/brick1/test1/

##### 3.分布复制卷

最少需要四台服务器才能创建，块服务器必须是复制的倍数，彼此相邻的块服务器成为彼此的复制

eg：

gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4

18节点：

[root@lvs ~]# ls /data0/gluster/data\_rd\_1/

testfile5  testfile7  testfile9

[root@lvs ~]# ls /data0/gluster/data\_rd\_2

root.tar.gz  testfile1  testfile10  testfile2  testfile3  testfile4  testfile6  testfile8

19节点：

[root@lvs ~]# ls /data0/gluster/data\_rd\_1/

testfile5  testfile7  testfile9

[root@lvs ~]# ls /data0/gluster/data\_rd\_2

root.tar.gz  testfile1  testfile10  testfile2  testfile3  testfile4  testfile6  testfile8

可以看到同一个节点下的两个brick构成哈希卷，文件哈希分配到两个brick上，而两个节点对应的brick上都是对方数据的完成副本，这个应该相当于raid10吧。

##### 4.条带卷

数据被分割成更小块分布到快服务器群中的不同条带区

分布减少了负载且更小的文件加速了存取的速度

没有数据冗余（仅在高并发环境中访问非常大的文件时才使用）

条带卷将一个文件分成两个部分存储在不同的brick下，同时拆分数据，没有冗余，不能坏盘，任何一个节点出现问题，所有文件都不能访问

eg:

gluster volume create test-volume3 stripe 2 transport tcp server1:/data/brick1/test1/ server2:/data/brick1/test1/

[root@lvs.17.sg1.test stripe]# pwd

/data0/stripe

[root@lvs.17.sg1.test stripe]# du -sh /root/root.tar.gz

140M    /root/root.tar.gz

[root@lvs.17.sg1.test stripe]# cp /root/root.tar.gz  .

在192.168.21.17上有一个140M的文件，复制到挂载了条待卷的目录下

18节点：

[root@lvs ~]# ls /data0/gluster/data3/

root.tar.gz

[root@lvs ~]# du -sh /data0/gluster/data3/root.tar.gz

70M     /data0/gluster/data3/root.tar.gz

19节点：

[root@lvs ~]# du -sh /data0/gluster/data3/root.tar.gz

70M     /data0/gluster/data3/root.tar.gz

#### 二、glusterfs服务端部署

初装配置

|  |  |  |
| --- | --- | --- |
| 主机名 | Ip地址 | 盘符 |
| gluster-node1 | 192.168.189.131 | /dev/sdb |
| gluster-node2 | 192.168.189.132 | /dev/sdb |
| localhost | 192.168.189.150 |  |

##### 1.安装常见依赖包

yum install -y vim net-tools ntpdate

##### 2.配置yum源，安装glusterfs相关包

yum install epel-release -y

yum install -y centos-release-gluster6.noarch

yum install glusterfs-server glusterfs glusterfs-fuse glusterfs-rdma -y

##### 3.启动glusterd服务并设置开机自启

systemctl start glusterd

systemctl enable glusterd

##### 4.设置主机名并写入hosts文件

hostnamectl set-hostname glusterfs-node1

su

hostnamectl set-hostname glusterfs-node2

su

[root@glusterfs-node1 ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.189.131 glusterfs-node1

192.168.189.132 glusterfs-node2

scp -rp /etc/hosts glusterfs-node2:/etc/

##### 5.两台分别关闭防火墙

systemctl stop firewalld

systemctl disable firewalld

setenforce 0

sed -i 's/enforcing/disabled/g' /etc/selinux/config

##### 6.配置ssh互信

[root@glusterfs-node1 ~]# ssh-keygen

[root@glusterfs-node1 ~]# ssh-copy-id glusterfs-node2

[root@glusterfs-node2 ~]# ssh-keygen

[root@glusterfs-node2 ~]# ssh-copy-id glusterfs-node1

##### 7.配置时间同步

###### 7.1更改当前时区

[root@glusterfs-node1 ~]# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

[root@glusterfs-node1 ~]# ntpdate cn.pool.ntp.org

[root@glusterfs-node2 ~]# cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

[root@glusterfs-node2 ~]# ntpdate cn.pool.ntp.org

###### 7.2 配置时间同步服务端

修改glusterfs-node1的/etc/’chrony.conf文件如下：

[root@glusterfs-node1 ~]# grep -v "#" /etc/chrony.conf

server cn.pool.ntp.org iburst

driftfile /var/lib/chrony/drift

makestep 1.0 3

rtcsync

allow 192.168.189.0/24

local stratum 10

logdir /var/log/chrony

###### 7.3 配置时间同步客户端

修改glusterfs-node2的/etc/’chrony.conf文件如下：

[root@glusterfs-node2 ~]# grep -v "#" /etc/chrony.conf

server 192.168.189.131 iburst

driftfile /var/lib/chrony/drift

makestep 1.0 3

rtcsync

logdir /var/log/chrony

###### 7.4 两台分别重启chrony服务并设置开机自启

Systemctl restart chronyd

Systemctl enable chronyd

###### 7.5 查看当前时间同步状态

[root@glusterfs-node1 ~]# chronyc sources -v

[root@glusterfs-node2 ~]# chronyc sources -v

##### 8. 添加节点

###### 8.1 节点一

[root@glusterfs-node1 ~]# gluster peer probe glusterfs-node2

peer probe: success.

[root@glusterfs-node1 ~]# gluster peer status

Number of Peers: 1

Hostname: glusterfs-node2

Uuid: 1131bc82-350e-4da0-9ec9-a4f25f768dc8

State: Peer in Cluster (Connected)

###### 8.2 节点二

[root@glusterfs-node2 ~]# gluster peer probe glusterfs-node1

##### 9. 创建卷，双复制卷

gluster v create fuzhi transport tcp replica 2 glusterfs-node1:/gluster/brick1/ glusterfs-node2:/gluster/brick1/

启动卷

gluster v start fuzhi

查看卷组状态

gluster v status

##### 10. 将盘符写入/etc/fstab

必须将gluster的brick所在的盘符设置为开机自动挂载

#### 三、glusterfs客户端安装并挂载

##### 1. 安装客户端相关包

yum -y install glusterfs glusterfs-fuse glusterfs-cli glusterfs-libs glusterfs-client-xlator

##### 2. 挂载卷组

**未配置hosts文件时，挂载会报错如下：**

[root@ceph-node1 ~]# mount.glusterfs 192.168.189.132:/replica2 /replica/

Mount failed. Please check the log file for more details.

日志路径如下：

/var/log/glusterfs/replica-.log

**日志输出如下：**

**DNS resolution failed on host glusterfs-node1**

**客户端必须配置hosts文件才可以挂载，该版本如下（不知道是该版本问题还是glusterfs通有问题）：**

[root@localhost ~]# rpm -qa | grep gluster

glusterfs-libs-6.10-1.el7.x86\_64

glusterfs-client-xlators-6.10-1.el7.x86\_64

glusterfs-cli-6.10-1.el7.x86\_64

centos-release-gluster6-1.0-1.el7.centos.noarch

glusterfs-6.10-1.el7.x86\_64

glusterfs-fuse-6.10-1.el7.x86\_64

[root@localhost ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.189.131 glusterfs-node1

192.168.189.132 glusterfs-node2

配置hosts后再次执行命令挂载正常

[root@localhost ~]# mount.glusterfs 192.168.189.132:/fuzhi /media/

[root@localhost ~]# df -h

Filesystem Size Used Avail Use% Mounted on

devtmpfs 475M 0 475M 0% /dev

tmpfs 487M 0 487M 0% /dev/shm

tmpfs 487M 7.7M 479M 2% /run

tmpfs 487M 0 487M 0% /sys/fs/cgroup

/dev/mapper/centos-root 17G 1.8G 16G 11% /

/dev/sda1 1014M 137M 878M 14% /boot

tmpfs 98M 0 98M 0% /run/user/0

glusterfs-node1:/fuzhi 5.0G 84M 5.0G 2% /media

#### 四、glusterfs常用命令

##### 1. 启停/查看glusterd服务

systemctl start glusterd

systemctl stop glusterd

systemctl status glusterd

##### 2. 开机自启glusterd服务

systemctl enable glusterd

systemctl list-unit-files | grep glusterd

##### 3. 为存储池添加/移除服务器节点

gluster peer probe gluster2 添加服务器节点

gluster peer detach gluster2 移除服务器节点 移除节点前必须将该节点上的brick删除

gluster peer status 查看所有节点的基本状态

##### 4、创建/启动/停止/删除卷

gluster volume start 卷名<volname>

gluster volume stop 卷名<volname>

gluster volume delete 卷名<volname> 删除卷之前必须先停止卷，最后可清空brick server节点对应目录下的内容

rm -rf /dis\_volume/brick

[root@glusterfs-node1 brick]# gluster v stop test-replica4

Stopping volume will make its data inaccessible. Do you want to continue? (y/n) y

volume stop: test-replica4: success

[root@glusterfs-node1 brick]# gluster v delete test-replica4

Deleting volume will erase all information about the volume. Do you want to continue? (y/n) y

volume delete: test-replica4: success

##### 5、删除brick

gluster volume remove-brick volume-name server1:/PATH server2:/PATH start

gluster volume remove-brick volume-name server1:/PATH server2:/PATH commit

gluster volume remove-brick volume-name server1:/PATH server2:/PATH status

[root@glusterfs-node1 brick]# gluster v remove-brick replica2 glusterfs-node1:/test-replica{3..4}/brick start

[root@glusterfs-node1 brick]# gluster v remove-brick replica2 glusterfs-node1:/test-replica{3..4}/brick status

Node Rebalanced-files size scanned failures skipped status run time in h:m:s

--------- ----------- ----------- ----------- ----------- ----------- ------------ --------------

localhost 0 0Bytes 9 0 0 in progress 0:00:10

The estimated time for rebalance to complete will be unavailable for the first 10 minutes.

[root@glusterfs-node1 brick]# gluster v remove-brick replica2 glusterfs-node1:/test-replica{3..4}/brick commit

如果是replica volume删除brick时必须削减副本数

如果是Distribute volume可以直接remove brick

[root@glusterfs-node1 brick]# gluster v remove-brick distribute1 glusterfs-node2:/gluster/brick start

Running remove-brick with cluster.force-migration enabled can result in data corruption. It is safer to disable this option so that files that receive writes during migration are not migrated.

Files that are not migrated can then be manually copied after the remove-brick commit operation.

Do you want to continue with your current cluster.force-migration settings? (y/n) y

volume remove-brick start: success

ID: ae22104c-5fbf-42b7-a06d-f4140cdbf58d

如果无法删除，执行如下命令,将gluster节点的配置文件删除

rm -rf /var/lib/glusterd/peers/\*

##将gluster配置全部删除重新配置

rm -rf /var/lib/glusterd

gluster peer status

**格式化要做gluster的盘符**

**mkfs.xfs /dev/sdb**

**创建挂载点，mkdir -p /data**

**将盘符挂载 mount /dev/sdb /data**

**在/etc/fstab里加入**

**/dev/sdb的UUID /data xfs defaults 0 0**

##### 客户端挂载gluster

1. 客户端以glusterfs方式挂载

创建挂载点

mkdir /datavol

mount -t glusterfs gluster1:/datavol /datavol

mount -t glusterfs <server>:/<volume> <mountdir>

节点:/卷名 挂载点

1. 客户端以nfs方式挂载

glusterfs默认没有打开nfs挂载方式，需要在服务端打开

服务端：

开放nfs

gluster volume set fuzhi nfs.disable off

卷名

gluster volume stop fuzhi

gluster volume start fuzhi

安装nfs-ganesha包

yum install -y nfs-ganesha nfs-ganesha-gluster

修改配置文件

[root@glusterfs-node1 /]# grep -Ev "#|^$" /etc/ganesha/ganesha.conf

EXPORT

{

Export\_Id=1 ;

Path = "/glu"; //指定nfs共享目录的位置，客户端挂载使用的就是服务端的该目录

Pseudo = "/glu-pseudo";

Disable\_ACL =True;

Protocols = "3","4";

Access\_Type = RW;

Squash = No\_root\_squash;

Sectype="sys";

Transports = "UDP","TCP";

FSAL { //定义的是准备导出的gluster的volume

Name = "GLUSTER"; //是应该导出卷的卷格式GLUSTER

Hostname="glusterfs-node1"; //是主机名 ，不同主机是不同的

Volume = "fuzhi"; //是gluster的卷名

}

}

重启nfs-ganesha服务

systemctl restart nfs-ganesha

客户端：

yum install -y nfs-utils

mkdir /nfs

查看nfs共享目录

showmount -e 192.168.189.131

开始挂载

mount.nfs 192.168.189.131:/glu /nfs/

##### 查看卷信息

* 1. 列出集群中的所有卷

gluster volume list

* 1. 查看集群中的卷信息

gluster volume info

* 1. 查看集群中的卷状态

gluster volume status

gluster volume status <volname>

1. 扩展卷

在gluster2上添加两块磁盘

先格式化mkfs.xfs /dev/sdc mkfs.xfs /dev/sdd

创建挂载点mkdir -p /data1 /data2

mount /dev/sdc /data1

mount /dev/sdd /data2

将挂载信息写入/etc/fstab文件中

云桌面三加一模式，第四个服务器可以使用如下方法添加至gluster集群里，添加的brick必须是replica的整数倍

gluster volume add-brick datavol gluster2:/data1/brick1 gluster2:/data2/brick1 force

**一个副本组的brick**默认不能在同一服务器上，如果处于同一服务器后面必须加上force参数才可以添加成功

##### brick管理

1、添加brick

gluster volume add-brick volume-name server1:/PATH server2:/PATH

2、移除brick

gluster volume remove-brick VOLNAME BRICK [start | status | commit]

3、替换brick

gluster volume replace-brick VOLNAME BRICKNEW-BRICK [start | pause | sbortstatus | commit]

2、手动同步数据

gluster volume volume-name reblance status

gluster volume volume-name reblance start

————————————————

##### 创建仲裁卷

如果是副本为3的仲裁器卷，其中第三个brick 充当仲裁器brick。 该配置具有防止发生裂脑的机制。

你可以使用下面的命令创建它：

# gluster volume create <VOLNAME> replica 3 arbiter 1 host1:brick1 host2:brick2 host3:brick3`

##### 重命名glusterfs集群名

要将群集 OpsClust 重命名为 OpsClust1，请键入：

**cluster opsclust /rename:opsclust1**

##### 防止脑裂

gluster volume set <volname> cluster.server-quorum-type none/server

 #gluster volume set all cluster.server-quorum-ratio <percentage%>

gluster volume set ksvd\_vol cluster.quorum-type auto

#### 五、限额quota

启用/禁用配额

gluster volume quota <VOLNAME> enable/disable

设置配额，此次的DIR为volume的相对路径

gluster volume quota <VOLNAME> limit-usage <DIR> <HARD\_LIMIT>

[root@glusterfs-node2 brick]# gluster volume quota replica2 limit-usage /test1/ 100

volume quota : success

[root@glusterfs-node2 brick]# gluster v quota replica2 list /test1

Path Hard-limit Soft-limit Used Available Soft-limit exceeded? Hard-limit exceeded?

-------------------------------------------------------------------------------------------------------------------------------

/test1 100Bytes 80%(80Bytes) 0Bytes 100Bytes No No

修改目录配额

[root@glusterfs-node2 brick]# gluster volume quota replica2 limit-usage /test1/ 100MB

volume quota : success

查看配额

[root@glusterfs-node2 ~]# gluster v quota replica2 list

查看某一个目录的配额

[root@glusterfs-node2 brick]# gluster v quota replica2 list /test1

Path Hard-limit Soft-limit Used Available Soft-limit exceeded? Hard-limit exceeded?

-------------------------------------------------------------------------------------------------------------------------------

/test1 100.0MB 80%(80.0MB) 0Bytes 100.0MB No No

开启限额quota

[root@glusterfs-node2 ~]# gluster v quota replica2 enable

volume quota : success

查看

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: e041e77e-0b24-4290-a451-ba14acaded92

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/glusterfs/replica/brick

Brick2: glusterfs-node2:/glusterfs/replica/brick

Options Reconfigured:

features.quota-deem-statfs: on

features.inode-quota: on

features.quota: on

cluster.quorum-type: auto

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

#### 六、常见故障处理

##### 1. 从一个brick迁移到另外一个brick

gluster volume replace-brick volume source-brick new-brick commit force

new-brick未被其他gluster使用并且和源brick大小一致

eg: gluster volume replace-brick replica2 192.168.189.132:/gluster1/brick1 192.168.189.132:/gluster/brick1 commit force

迁移完成后，源brick就不会再存储新写入的数据

##### 2. 双复制一个节点的磁盘损坏

节点：gluster1、gluster2

gluster1节点的brick硬盘损坏

移除gluster1的brick

1）初始状态

[root@gluster2 ~]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

------------------------------------------------------------------------------

Brick gluster1:/data/brick1 49153 0 Y 3047

Brick gluster2:/data/brick1 49155 0 Y 1617

Self-heal Daemon on localhost N/A N/A Y 1634

Self-heal Daemon on gluster1 N/A N/A Y 3064

Task Status of Volume fuzhi

------------------------------------------------------------------------------

There are no active volume tasks

2）移除brick

[root@gluster2 ~]# gluster volume remove-brick fuzhi replica 1 gluster2:/data/brick1/ force

3）查看当前状态

[root@gluster2 ~]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

------------------------------------------------------------------------------

Brick gluster1:/data/brick1 49153 0 Y 3047

Task Status of Volume fuzhi

------------------------------------------------------------------------------

There are no active volume tasks

4）添加新的brick

[root@gluster2 ~]# gluster volume add-brick fuzhi replica 2 gluster2:/data1/brick1/

Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3 to avoid this. See: http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/.

Do you still want to continue?

(y/n) y

volume add-brick: failed: /data1/brick1 is already part of a volume

brick添加失败，该brick已是卷的一部分，对于硬盘原有做过gluster，该硬盘的brick1下仍存在原有的brick信息，将原有信息删除

5）删除原有brick相关信息

[root@gluster2 ~]# rm -rf /data1/brick1/

6) 重新添加brick

[root@gluster2 data1]# gluster volume add-brick fuzhi replica 2 gluster2:/data1/brick1/

Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3 to avoid this. See: http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/.

Do you still want to continue?

(y/n) y

volume add-brick: success

7）查看当前状态

[root@gluster2 data1]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

------------------------------------------------------------------------------

Brick gluster1:/data/brick1 49153 0 Y 3047

Brick gluster2:/data1/brick1 49155 0 Y 2063

Self-heal Daemon on localhost N/A N/A Y 2080

Self-heal Daemon on gluster1 N/A N/A Y 3493

Task Status of Volume fuzhi

------------------------------------------------------------------------------

There are no active volume tasks

##### 3. 将二复制转为三复制

当前状态

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

节点glusterfs-node2新增一块和已存在的两块brick大小一致的硬盘

Vmware热添加硬盘

echo '- - -' >/sys/class/scsi\_host/host0/scan

echo '- - -' >/sys/class/scsi\_host/host1/scan

echo '- - -' >/sys/class/scsi\_host/host2/scan

lsblk查看，多了一块sdc硬盘

[root@glusterfs-node2 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 20G 0 disk

├─sda1 8:1 0 1G 0 part /boot

└─sda2 8:2 0 19G 0 part

├─centos-root 253:0 0 17G 0 lvm /

└─centos-swap 253:1 0 2G 0 lvm [SWAP]

sdb 8:16 0 5G 0 disk /gluster

sdc 8:32 0 5G 0 disk

sr0 11:0 1 973M 0 rom

进行格式化并挂载

mkfs.xfs /dev/sdc

mkdir /gluster1

mount /dev/sdc /gluster1

mkdir -p /gluster1/brick1

添加brick到原有的二复制卷，并由二复制更改为三复制

gluster v add-brick replica2 replica 3 192.168.189.132:/gluster1/brick1/

卷名 三复制

更改完以后的gluster状态

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Brick3: 192.168.189.132:/gluster1/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

##### 4. 将三复制转为二复制

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Brick3: 192.168.189.132:/gluster1/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

移除一个brick并从三复制更改为二复制

gluster v remove-brick replica2 replica 2 192.168.189.132:/gluster1/brick1 force

卷名 二复制

更改以后的状态

[root@glusterfs-node2 ~]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

##### 5. 更改卷模式（从复制卷更改为分布式卷）

原有模式为二复制

[root@glusterfs-node2 brick1]# gluster v info

Volume Name: replica2

Type: Replicate

Volume ID: 64d9b9e6-e9f1-45d4-8258-3e8bb00c3f00

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: 192.168.189.132:/gluster/brick1

Options Reconfigured:

performance.client-io-threads: off

nfs.disable: on

transport.address-family: inet

删除其中一个brick

gluster volume remove-brick replica2 replica 1 glusterfs-node2:/gluster/brick1 force

停止卷volume

gluster volume stop replica2

gluster volume delete replica2

再创建volume

gluster v create replica2 glusterfs-node1:/gluster/brick1/ glusterfs-node2:/gluster/brick1/ force

更改以后的模式为分布式

[root@glusterfs-node2 brick1]# gluster v info

Volume Name: replica2

Type: Distribute

Volume ID: 003cc5fc-d753-4666-8222-b10d12c865cf

Status: Created

Snapshot Count: 0

Number of Bricks: 2

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/gluster/brick1

Brick2: glusterfs-node2:/gluster/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

启动volume

gluster v start replica2

更改为分布式后，所有

##### 6. 对volume扩容

1. 初始状态

[root@gluster2 data1]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

------------------------------------------------------------------------------

Brick gluster1:/data/brick1 49153 0 Y 3047

Brick gluster2:/data1/brick1 49155 0 Y 2063

Self-heal Daemon on localhost N/A N/A Y 2080

Self-heal Daemon on gluster1 N/A N/A Y 3493

Task Status of Volume fuzhi

------------------------------------------------------------------------------

There are no active volume tasks

1. 扩展volume

[root@gluster2 data1]# gluster volume add-brick fuzhi replica 2 gluster2:/data/brick1 gluster2:/data2/brick1 force

volume add-brick: success

[root@glusterfs-node1 test-replica1]# gluster v add-brick replica2 glusterfs-node1:/test-replica{3..4}/brick

volume add-brick: success

force：因为两个brick属于同一个节点，所有必须加上force

1. 查看当前状态

[root@gluster2 data1]# gluster volume info

Volume Name: fuzhi

Type: Distributed-Replicate

Volume ID: e1ccb13f-8aad-4aaf-897f-2c9915b70686

Status: Started

Snapshot Count: 0

Number of Bricks: 2 x 2 = 4

Transport-type: tcp

Bricks:

Brick1: gluster1:/data/brick1

Brick2: gluster2:/data1/brick1

Brick3: gluster2:/data/brick1

Brick4: gluster2:/data2/brick1

Options Reconfigured:

cluster.granular-entry-heal: on

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

[root@gluster2 data1]# gluster volume status

Status of volume: fuzhi

Gluster process TCP Port RDMA Port Online Pid

------------------------------------------------------------------------------

Brick gluster1:/data/brick1 49153 0 Y 3047

Brick gluster2:/data1/brick1 49155 0 Y 2063

Brick gluster2:/data/brick1 49156 0 Y 2489

Brick gluster2:/data2/brick1 49158 0 Y 2505

Self-heal Daemon on localhost N/A N/A Y 2080

Self-heal Daemon on gluster1 N/A N/A Y 3493

Task Status of Volume fuzhi

------------------------------------------------------------------------------

There are no active volume tasks

##### 7. 重新均衡卷

哈希volume才会用的均衡

不迁移数据：

# gluster volume rebalance <VOLNAME> lay-out start

只修复目录的哈希分布，并不会实际迁移文件，此时新增文件就可以存储到新增brick上

开始迁移数据

# volume rebalance <VOLNAME> {{fix-layout start} | {start [force]|stop|status}}

##### 8. 系统扩展维护

开启/关闭系统配额：

# gluster volume quota <VOLNAME> enable | disable

设置目录配额：

# gluster volume quota <VOLNAME> limit-usage <DIR> <VALUE>

查看配额：

# gluster volume quota <VOLNAME> list [<DIR>]

地域复制（geo-replication）：

# gluster volume geo-replication <MASTER> <SLAVE> start | status | stop

IO信息查看：

# gluster volume profile <VOLNAME> start | info | stop

Top监控：

Top命令允许你查看Brick的性能，例如：read,write, file open calls, file read calls, file write calls, directory opencalls, and directory real calls。所有的查看都可以设置 top数，默认100。

查看打开的 fd：

# gluster volume top <VOLNAME> open[brick <BRICK>] [list-cnt <COUNT>]

其中，open可以替换为read, write, opendir, readdir等。

查看每个 Brick 的读性能：

# gluster volume top <VOLNAME> read-perf [bs <BLOCK-SIZE> count <COUNT>] [brick <BRICK>] [list-cnt <COUNT>]

其中，read-perf可以替换为write-perf等。

##### 9. GlusterFS优化

# 设置 cache 大小, 默认32MB

gluster volume set senyintvolume performance.cache-size 4GB

# 设置 io 线程, 太大会导致进程崩溃

gluster volume set senyintvolume performance.io-thread-count 16

# 设置 网络检测时间, 默认42s

gluster volume set senyintvolume network.ping-timeout 10

# 设置 写缓冲区的大小, 默认1M

gluster volume set senyintvolume performance.write-behind-window-size 1024MB

# 开启 指定 volume 的配额，不使用

gluster volume quota k8s-volume enable

# 限制 指定 volume 的配额，不使用

gluster volume quota k8s-volume limit-usage / 1TB

##### 常见问题

创建gluster卷时报错

volume create: replica2: failed: /gluster/brick1 is already part of a volume

各节点都需要执行

删除对应brick下的 .glusterfs目录

setfattr -x trusted.glusterfs.volume-id brick

setfattr -x trusted.gfid brick

再次执行创建操作即可

#### 七、glusterfs相关日志

相关日志，在/var/log/glusterfs/目录下，可根据需要查看；

如/var/log/glusterfs/brick/下是各brick创建的日志；

如/var/log/glusterfs/cmd\_history.log是命令执行记录日志；

如/var/log/glusterfs/glusterd.log是glusterd守护进程日志。

#### 八、glusterfs脑裂处理

脑裂分为三种

1. 数据脑裂：文件中的数据在副本组的brick中不同
2. 元数据脑裂：brick中元数据不同
3. GFID脑裂：副本brick上的文件GFID不同，或者副本上的文件类型不同，文件类型不同无法修复，GFID可以修复，**GFID脑裂对外表现为目录脑裂**

列出所有需要修复的文件，glusterfs有自动修复脑裂的机制，列出的文件或目录不一定都需要手动修复

# gluster volume heal test info

Brick \<hostname:brickpath-b1>

<gfid:aaca219f-0e25-4576-8689-3bfd93ca70c2> - Is in split-brain

<gfid:39f301ae-4038-48c2-a889-7dac143e82dd> - Is in split-brain

<gfid:c3c94de2-232d-4083-b534-5da17fc476ac> - Is in split-brain

<gfid:6dc78b20-7eb6-49a3-8edb-087b90142246>

Number of entries: 4

Brick <hostname:brickpath-b2>

/dir/file2

/dir/file1 - Is in split-brain

/dir - Is in split-brain

/dir/file3

/file4 - Is in split-brain

/dir/a

Number of entries: 6

输出分析，

A） brickpath-b1 中，需要修复四个条目：

* 带有 GFID：6dc78b20-7EB6-49A3-8EDB-087B90142246 的文件需要愈合
* “AACA219F-0E25-4576-8689-3BFD93CA70C2”、“39F301AE-4038-48C2-A889-7DAC143E82DD”和“C3C94DE2-232D-4083-B534-5DA17FC476AC”属于裂脑

B） brickpath-b2中，需要修复六个条目：

* “a”、“file2”和“file3”需要修复
* “file1”、“file4”和“/dir”在裂脑中

因为/dir/a文件GFID脑裂，导致/dir处于脑裂状态

列出需要手动修复的文件和目录

# gluster volume heal test info split-brain

Brick <hostname:brickpath-b1>

<gfid:aaca219f-0e25-4576-8689-3bfd93ca70c2>

<gfid:39f301ae-4038-48c2-a889-7dac143e82dd>

<gfid:c3c94de2-232d-4083-b534-5da17fc476ac>

Number of entries in split-brain: 3

Brick <hostname:brickpath-b2>

/dir/file1

/dir

/file4

Number of entries in split-brain: 3

##### 1、使用gluster cli修复数据/元数据脑裂(server)

###### 选择较大的文件作为源

gluster volume heal <VOLNAME> split-brain bigger-file <FILE>

###### 选择最新的mtime作为源

gluster volume heal <VOLNAME> split-brain latest-mtime <FILE>

###### 选择副本中的一个brick作为特定文件的源

gluster volume heal <VOLNAME> split-brain source-brick <HOSTNAME:BRICKNAME> <FILE>

<FILE>可以以指定gfid的方式标识

gluster volume heal test split-brain source-brick test-host:/test/b1 gfid:c3c94de2-232d-4083-b534-5da17fc476ac

###### 选择副本中的一个brick作为所有文件的源

gluster volume heal <VOLNAME> split-brain source-brick <HOSTNAME:BRICKNAME>

##### 2、使用gluster cli修复GFID脑裂(server)

列出脑裂，该文件的父目录处于脑裂中，说明就是GFID脑裂

# gluster volume heal testvol info

Brick 10.70.47.45:/bricks/brick2/b0

/f5

/ - Is in split-brain

Status: Connected

Number of entries: 2

Brick 10.70.47.144:/bricks/brick2/b1

/f5

/ - Is in split-brain

Status: Connected

Number of entries: 2

对于GFID脑裂，它是父目录，显示处于脑裂中

使用如下命令获取文件是否处于GFID脑裂

# getfattr -d -e hex -m. <path-to-file>

在brick b0上

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

getfattr: Removing leading '/' from absolute path names

file: bricks/brick2/b0/f5

security.selinux=0x73797374656d5f753a6f626a6563745f723a676c7573746572645f627269636b5f743a733000

trusted.afr.testvol-client-1=0x000000020000000100000000

trusted.afr.dirty=0x000000000000000000000000

**trusted.gfid**=0xce0a9956928e40afb78e95f78defd64f

trusted.gfid2path.9cde09916eabc845=0x30303030303030302d303030302d303030302d303030302d3030303030303030303030312f6635

在brick b1上

# getfattr -d -m . -e hex /bricks/brick2/b1/f5

getfattr: Removing leading '/' from absolute path names

file: bricks/brick2/b1/f5

security.selinux=0x73797374656d5f753a6f626a6563745f723a676c7573746572645f627269636b5f743a733000

trusted.afr.testvol-client-0=0x000000020000000100000000

trusted.afr.dirty=0x000000000000000000000000

**trusted.gfid**=0x9563544118653550e888ab38c232e0c

trusted.gfid2path.9cde09916eabc845=0x30303030303030302d303030302d303030302d303030302d3030303030303030303030312f6635

可以看到两个brick中gfid不同

###### 选择较大的文件作为源

# gluster volume heal VOLNAME split-brain bigger-file FILE

先使用stat <file>查看文件大小

然后再选择较大的文件作为源修复文件的GFID脑裂

# gluster volume heal testvol split-brain bigger-file /f5

GFID split-brain resolved for file /f5

修复完成后，两个brick上的GFID必须与较大尺寸的GFID相同，修复完成后可以使用如下命令查看GFID使用相同

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

###### 选择最新的mtime作为源

# gluster volume heal VOLNAME split-brain latest-mtime FILE

修复完成后，两个brick上的GFID必须与较大尺寸的GFID相同，修复完成后可以使用如下命令查看GFID使用相同

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

###### 选择副本中的一个brick作为特定文件的源

# gluster volume heal VOLNAME split-brain source-brick HOSTNAME:export-directory-absolute-path FILE

修复完成后，两个brick上的GFID必须与较大尺寸的GFID相同，修复完成后可以使用如下命令查看GFID使用相同

# getfattr -d -m . -e hex /bricks/brick2/b0/f5

###### 注意

* 不能将文件的 GFID 用作任何 CLI 选项的参数来解决 GFID 裂脑问题。它应该是被视为源的文件的绝对路径。
* 使用brick作为源选项，无法一次性解决所有 GFID 裂脑，因为在解析数据或元数据裂脑时，无需在 CLI 中指定任何文件路径。对于 GFID 裂脑中的每个文件，使用要使用的策略运行 CLI。
* 使用带有“分布式复制”卷中的“brick块”选项的 CLI 解析目录 GFID 裂脑需要在处于此状态的所有子卷上显式完成。由于目录会在所有brick上创建，因此使用一个特定的brick作为目录 GFID 裂脑的源可以修复该特定子卷的目录。源brick的选择方式应使修复后所有子卷的所有brick都具有相同的 GFID。
* 如前所述，无法使用CLI解决文件系统类型不匹配的问题

##### 3、从挂载点（client）修复脑裂

提供了一组 getfattr 和 setfattr 命令来检测文件的数据和元数据裂脑状态，并从挂载点解析裂脑（如果有）。

# gluster v heal test info split-brain

Brick test-host:/test/b0/

/file100

/dir

Number of entries in split-brain: 2

Brick test-host:/test/b1/

/file100

/dir

Number of entries in split-brain: 2

Brick test-host:/test/b2/

/file99

<gfid:5399a8d1-aee9-4653-bb7f-606df02b3696>

Number of entries in split-brain: 2

Brick test-host:/test/b3/

<gfid:05c4b283-af58-48ed-999e-4d706c7b97d5>

<gfid:5399a8d1-aee9-4653-bb7f-606df02b3696>

Number of entries in split-brain: 2

使用如下命令了解处于哪种脑裂状态

getfattr -n replica.split-brain-status <path-to-file>

eg:

处于元数据脑裂

# getfattr -n replica.split-brain-status file100

file: file100

replica.split-brain-status="data-split-brain:no metadata-split-brain:yes Choices:test-client-0,test-client-1"

处于数据脑裂

# getfattr -n replica.split-brain-status file99

file: file99

replica.split-brain-status="data-split-brain:yes metadata-split-brain:yes Choices:test-client-2,test-client-3"

元数据和数据都处于脑裂中

# getfattr -n replica.split-brain-status file99

file: file99

replica.split-brain-status="data-split-brain:yes metadata-split-brain:yes Choices:test-client-2,test-client-3"

dir不在数据或者元数据脑裂下，

# getfattr -n replica.split-brain-status dir

file: dir

replica.split-brain-status="The file is not under data or metadata split-brain"

###### 从client解决数据和元数据脑裂

尝试在挂载点上对脑裂的文件进行操作（例如 cat、getfattr 等），会出现输入/输出错误。为了使用户能够分析此类文件，提供了一个 setfattr 命令。

# setfattr -n replica.split-brain-choice -v "choiceX" <path-to-file>

使用此命令，可以选择一个特定的块来访问裂脑中的文件。

eg：

1、“file1”位于数据脑裂中。尝试从文件中读取会产生输入/输出错误。

# cat file1

cat: file1: Input/output error

为file1提供的服务端选择有test-cilent-1和test-client-2

将test-client-2设置为file1的脑裂选择源，将从test-client-2的brick中读取文件

# setfattr -n replica.split-brain-choice -v test-client-2 file1

然后再cat文件

cat file1

xyz123

要撤销已设置的脑裂选择，可以使用“none”作为setfattr的扩展属性的值

# setfattr -n replica.split-brain-choice -v none file1

再cat文件，就会和以前一样，输出Input/output error

# cat file

cat: file1: Input/output error

如果想解决脑裂问题，应该设置源brick

# setfattr -n replica.split-brain-heal-finalize -v <heal-choice> <path-to-file>

eg：

# setfattr -n replica.split-brain-heal-finalize -v test-client-2 file1

上述过程可以解决所有文件的数据/元数据脑裂

###### 注意：

1、如果禁用了“fopen-keep-cache”保险丝装载选项，则每次在选择新副本之前都需要使 inode 失效。拆分大脑选择检查文件。这可以通过使用来完成：

# sefattr -n inode-invalidate -v 0 <path-to-file>

2、上面从client修复脑裂的过程不适合nfs挂载，因为nfs不提供xattrs支持

##### 4、设置glustefs，让gluseterfs自动修复脑裂

基于CLI和client的方法需要人工干预，有一个卷设置，当设置为各种可用策略之一时，无需用户干预即可自动恢复脑裂，默认被禁用。

cluster.favorite-child-policy

查看help

[root@glusterfs-node2 brick]# gluster v set help | grep -A3 cluster.favorite-child-policy

Option: cluster.favorite-child-policy

Default Value: none

Description: This option can be used to automatically resolve split-brains using various policies without user intervention.

"size" picks the file with the biggest size as the source. "ctime" and "mtime" pick the file with the latest ctime and mtime respectively as the source. "majority" picks a file with identical mtime and size in more than half the number of bricks in the replica.a

设置glusterfs以最新mtime自动修复脑裂

[root@glusterfs-node2 ~]# gluster v set replica2 cluster.favorite-child-policy mtime

volume set: success

[root@dockernode1 /]# gluster v info

Options Reconfigured:

nfs.disable: on #关闭nfs挂载

storage.fips-mode-rchecksum: on

transport.address-family: inet

cluster.quorum-type: auto #quorom-type为auto，正常brick必须满足半数以上才可以使用

none|auto|fixed,none:关闭，auto：半数以上，fixed：大于等于cluster.quorum-count设置的数量

cluster.favorite-child-policy: mtime #以最新的mtime自动修复脑裂

diagnostics.latency-measurement: on

diagnostics.count-fop-hits: on

##### 5、快速入门：

1. 获取裂脑中文件的路径：

它可以通过  
a） 命令获得。  
b） 确定从客户端执行的文件操作不断失败并出现输入/输出错误的文件。gluster volume heal info split-brain

1. 关闭从装入点打开此文件的应用程序。 对于虚拟机，需要关闭它们的电源。
2. 确定正确的副本：

这是通过观察文件的 afr 更改日志扩展属性来完成的 使用 getfattr 命令的砖块;然后确定裂脑的类型 （数据裂脑、元数据裂脑、条目裂脑或裂脑由于 GFID-不匹配）;最后确定哪个砖块包含“好副本” 的文件。  
.  
也可能一个砖可能包含正确的数据，而 其他可能包含正确的元数据。getfattr -d -m . -e hex <file-path-on-brick>

1. 重置包含 使用 setfattr 命令的文件数据/元数据的“错误副本”。

setfattr -n <attribute-name> -v <attribute-value> <file-path-on-brick>

1. 通过从客户端执行查找来触发对文件的自我修复：

ls -l <file-path-on-gluster-mount>

#### 九、深入探究glusterfs

|  |  |  |
| --- | --- | --- |
| glusterfs-node1 |  |  |
| glusterfs-node2 |  |  |
| glusterfs-client |  |  |

##### 双复制集群单一节点故障分析

双复制机器，当client和server端已建立连接后

任一节点或者双节点glusterd服务stop(glusterd被kill掉)不影响client端

任一节点的glusterfsd进程被kill掉，有一半几率会出问题

[root@glusterfs-node1 brick]# netstat -tunlp | grep "gluster"

tcp 0 0 0.0.0.0:49152 0.0.0.0:\* LISTEN 1101/glusterfsd

tcp 0 0 0.0.0.0:49153 0.0.0.0:\* LISTEN 4480/glusterfsd

tcp 0 0 0.0.0.0:49154 0.0.0.0:\* LISTEN 1116/glusterfsd

tcp 0 0 0.0.0.0:24007 0.0.0.0:\* LISTEN 4468/glusterd

[root@glusterfs-node2 ~]# netstat -tunlp | grep "gluster"

tcp 0 0 0.0.0.0:49152 0.0.0.0:\* LISTEN 1001/glusterfsd

tcp 0 0 0.0.0.0:49153 0.0.0.0:\* LISTEN 4228/glusterfsd

tcp 0 0 0.0.0.0:49154 0.0.0.0:\* LISTEN 1034/glusterfsd

tcp 2 0 0.0.0.0:24007 0.0.0.0:\* LISTEN 4216/glusterd

[root@ceph-node1 ~]# netstat -tunp | grep "gluster"

tcp 0 0 192.168.189.150:49147 192.168.189.131:24007 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49129 192.168.189.132:24007 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49151 192.168.189.131:24007 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49142 192.168.189.131:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49137 192.168.189.132:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49138 192.168.189.131:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49144 192.168.189.131:49153 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49143 192.168.189.132:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49130 192.168.189.132:49153 ESTABLISHED 1904/glusterfs

当kill掉glusterfs-node2的49153端口的进程时，查看glusterfs-client

[root@ceph-node1 ~]# !netstat

netstat -tunp

Active Internet connections (w/o servers)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 68 192.168.189.150:22 192.168.189.1:64938 ESTABLISHED 1195/sshd: root@pts

tcp 0 0 192.168.189.150:49147 192.168.189.131:24007 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49150 192.168.189.132:24007 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49151 192.168.189.131:24007 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49142 192.168.189.131:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49137 192.168.189.132:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49138 192.168.189.131:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49144 192.168.189.131:49153 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49143 192.168.189.132:49154 ESTABLISHED 1350/glusterfs

此时客户端还可以继续使用49153端口的replica2 volume

当kill掉glusterfs-node1的49153端口的进程时，查看glusterfs-client

[root@ceph-node1 ~]# netstat -tunp | grep "gluster"

tcp 0 0 192.168.189.150:49147 192.168.189.131:24007 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49129 192.168.189.132:24007 ESTABLISHED 1904/glusterfs

tcp 0 0 192.168.189.150:49151 192.168.189.131:24007 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49142 192.168.189.131:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49137 192.168.189.132:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49138 192.168.189.131:49152 ESTABLISHED 1409/glusterfs

tcp 0 0 192.168.189.150:49143 192.168.189.132:49154 ESTABLISHED 1350/glusterfs

tcp 0 0 192.168.189.150:49130 192.168.189.132:49153 ESTABLISHED 1904/glusterfs

此时glusterfs-client访问replica2 volume的挂载目录

[root@ceph-node1 ~]# cd /replica

-bash: cd: /replica: Transport endpoint is not connected

当server端为双复制的时候，且cluster.quorum-type: auto时，一个节点down机，有一半的可能导致client无法使用，与挂载时写的哪个serverIP无关系

当glusterfs-node1节点恢复正常后，client会自动恢复正常，无需其他操作

##### 2、glusterfs是否支持多副本？

[root@glusterfs-node1 brick]# gluster v create test-replica4 replica 4 glusterfs-node1:/test-replica1/brick/ glusterfs-node1:/test-replica2/brick/ glusterfs-node1:/test-replica3/brick/ glusterfs-node1:/test-re

plica4/brick/ forcevolume create: test-replica4: success: please start the volume to access data

[root@glusterfs-node1 brick]# gluster v status test-replica4

Volume test-replica4 is not started

启动test-replica4 volume

[root@glusterfs-node1 brick]# gluster v start test-replica4

volume start: test-replica4: success

[root@glusterfs-node1 brick]# gluster v status test-replica4

Status of volume: test-replica4

Gluster process TCP Port RDMA Port Online Pid

------------------------------------------------------------------------------

Brick glusterfs-node1:/test-replica1/brick 49155 0 Y 5202

Brick glusterfs-node1:/test-replica2/brick 49156 0 Y 5222

Brick glusterfs-node1:/test-replica3/brick 49157 0 Y 5242

Brick glusterfs-node1:/test-replica4/brick 49158 0 Y 5262

Self-heal Daemon on localhost N/A N/A Y 5283

Self-heal Daemon on glusterfs-node2 N/A N/A Y 4466

Task Status of Volume test-replica4

------------------------------------------------------------------------------

There are no active volume tasks

[root@glusterfs-node1 brick]# gluster v info test-replica4

Volume Name: test-replica4

Type: Replicate

Volume ID: dd9cc607-8ef9-4c6b-ba63-492fadadd084

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 4 = 4

Transport-type: tcp

Bricks:

Brick1: glusterfs-node1:/test-replica1/brick

Brick2: glusterfs-node1:/test-replica2/brick

Brick3: glusterfs-node1:/test-replica3/brick

Brick4: glusterfs-node1:/test-replica4/brick

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

可以看到glusterfs支持多副本（3副本以上）

##### 3、glusterfs rebalance volume

当add new brick后，新的brick里面都会创建已存在的目录，但是内容为空

当新增文件时，glusterfs会根据各个brick的剩余空间大小来决定写到那个brick里

当新增brick后，新增brick下会同步创建以前存在的目录，但目录下为空

新增brick以前创建的旧目录下新增文件默认不会在新brick下存储

volume根下的文件也不会在新的brick下存储

只有新创建的目录及新创建目录下的文件才会在新的brick下存储

###### 1）修复目录哈希分布，不会迁移数据，可以让旧目录下的文件存储在新brick下

[root@glusterfs-node2 kylin]# gluster v rebalance replica2 fix-layout start

volume rebalance: replica2: success: Rebalance on replica2 has been started successfully. Use rebalance status command to check status of the rebalance process.

ID: e3116dbe-60b9-4867-8f32-8f4a37d6c23f

当rebalance后在旧目录下或者volume根目录下创建文件，会在新的brick下存储

###### 2）均衡数据，会把新旧brick下的数据进行迁移均衡

[root@glusterfs-node2 brick2]# gluster v rebalance replica2 start

volume rebalance: replica2: success: Rebalance on replica2 has been started successfully. Use rebalance status command to check status of the rebalance process.

ID: 7f2832d8-8f65-4c90-8fbe-fbbd65faf89d

###### 3）设置迁移速度

gluster volume set <VOLUME> rebal-throttle lazy|normal|aggressive

默认normal

Lazy:慢速模式，较少线程迁移

Normal：正常模式线程数量适中

Aggressive：激进模式，较多线程迁移

###### 注意

1. 如果有文件损坏，先修复
2. 均衡前，确认集群没有自修复在进行否则会影响数据正确率和迁移效率
3. 先执行fix-layout再进行数据迁移，可以提高迁移效率
4. 如果可以，停止客户端使用，再迁移，可以提高迁移效率
5. 迁移过程中，使用status时刻查看迁移状态
6. 当集群较大时，可能会出现某个节点均衡失败的问题，一般重新开始执行均衡即可
7. 如果迁移对程序影响较大，可以只执行fix-layout，只修复目录哈希分布，不会实际迁移数据，此时新文件会存储在新节点或者brick上

#### 十、FAQ

##### 1、volume add-brick: failed扩展或者创建volume时

[root@glusterfs-node1 test-replica1]# gluster v add-brick replica2 glusterfs-node1:/test-replica1/ glusterfs-node1:/test-replica2

volume add-brick: failed: The brick glusterfs-node1:/test-replica1 is a mount point. Please create a sub-directory under the mount point and use that as the brick directory. Or use 'force' at the end of the command if you want to override this behavior.

直接使用disk挂载点创建会报错

在挂载点下创建目录再创建volume

[root@glusterfs-node1 test-replica1]# mkdir /test-replica{1..2}/brick

[root@glusterfs-node1 test-replica1]# gluster v add-brick replica2 glusterfs-node1:/test-replica1/brick glusterfs-node1:/test-replica2/brick

volume add-brick: success

##### 2、volume remove-brick start: failed缩容或者删除volume时

[root@glusterfs-node1 brick]# gluster v remove-brick test-replica4 glusterfs-node4:/test-replica4/brick/ start

Running remove-brick with cluster.force-migration enabled can result in data corruption. It is safer to disable this option so that files that receive writes during migration are not migrated.

Files that are not migrated can then be manually copied after the remove-brick commit operation.

Do you want to continue with your current cluster.force-migration settings? (y/n) y

volume remove-brick start: failed: Removing bricks from replicate configuration is not allowed without reducing replica count explicitly.

当remove brick时如果volume为replica volume，remove的brick必须为volume的倍速，否则报错