

3.4.3. graph 和 volfile

这里分享一下什么是 volfile 和 graph,其实在前面的一节内容中,这里也有使用到 volfile 文件,就是在/var/lib/glusterd/vols 下面的文件中,如下所示。

```
1. [root@gfs01 ~]# ls /var/lib/glusterd/vols/test-replica/  
2. bricks                quota.conf  
                           test-replica.gfproxyd.vol  
3. cksum                 snapd.info  
                           test-replica-shd.vol  
4. info                  test-replica.192.168.0.110.glusterfs-test-replica.vol  
                           test-replica.tcp-fuse.vol  
5. node_state.info       test-replica.192.168.0.111.glusterfs-test-replica.vol  
                           trusted-test-replica.tcp-fuse.vol  
6. quota.cksum           test-replica.192.168.0.112.glusterfs-test-replica.vol  
                           trusted-test-replica.tcp-gfproxy-fuse.vol  
7.  
8. [root@gfs01 ~]# ls /var/lib/glusterd/vols/test-replica/test-replica.192.168.0.110.glusterfs-test-replica.vol  
9. /var/lib/glusterd/vols/test-replica/test-replica.192.168.0.110.glusterfs-test-replica.vol
```

这里的 vol 文件就是一个 volfile 了,那么这里为什么 glusterfs 官方不使用 xml,json 或者 yaml 文件呢?因为 glusterfs 是项目在 2005 就开始了,那时候现在主流的配置文件格式这些还没有,或者还不是很出名,因此当时就选择 volfile 了,也一直保留了下来。另外因为这个配置文件的格式问题,也不算是集群的一些影响到性能或者代码层次结构的,因此虽然没有用到现在主流的方式,但是并不影响真正使用。

那么这里 volfile 中记录的其实就是 graph,也就是不同的 xlator 功能结构的组织结构顺序,下面可以简单看看这个文件来感受一下。

```
1. [root@gfs01 ~]# cat /var/lib/glusterd/vols/test-replica/test-replica.192.168.0.110.glusterfs-test-replica.vol
```

```

2. volume test-replica-posix
3.     type storage/posix
4.     option shared-brick-count 1
5.     option fips-mode-rchecksum on
6.     option volume-id 4568c063-5b75-4304-98f6-21f3955cc138

7.     option directory /glusterfs/test-replica
8. end-volume
9.
10. volume test-replica-trash
11.     type features/trash
12.     option trash-internal-op off
13.     option brick-path /glusterfs/test-replica
14.     option trash-dir .trashcan
15.     subvolumes test-replica-posix
16. end-volume
17.
18. ....
19.
20. volume /glusterfs/test-replica
21.     type debug/io-stats
22.     option global-threading off
23.     option count-fop-hits off
24.     option latency-measurement off
25.     option threads 16
26.     option log-level DEBUG
27.     option volume-id 4568c063-5b75-4304-98f6-21f3955cc138

28.     option unique-id /glusterfs/test-replica
29.     subvolumes test-replica-quota
30. end-volume
31.
32. volume test-replica-server
33.     type protocol/server
34.     option transport.listen-backlog 1024
35.     option transport.socket.keepalive-count 9
36.     option transport.socket.keepalive-interval 2
37.     option transport.socket.keepalive-time 20
38.     option transport.socket.ssl-enabled off
39.     option transport.socket.keepalive 1
40.     option auth.addr./glusterfs/test-replica.allow *
41.     option auth-path /glusterfs/test-replica
42.     option auth.login.fad65c15-ff26-4465-914a-eb83bb332fff
        f.password b2784284-715c-4e46-a1ca-97b86eb7fa1a

```

```
43.      option auth.login./glusterfs/test-replica.allow fad65
        c15-ff26-4465-914a-eb83bb332fff
44.      option transport.address-family inet
45.      option transport-type tcp
46.      subvolumes /glusterfs/test-replica
47. end-volume
```

这里有几个地方值得注意的，首先是这个文件的格式，每一个 xlator 层次都是用 volume 然后 end-volume 来结束的，这样看起来和 xml 中的格式有点类似。接着这里还要留意一下 subvolumes 的内容，subvolumes 是什么内容呢？简单理解就是记录下一个 xlator 是哪个，在 volfile 文件中，这里要注意到，文件的顺序是从下往上的，这一点就可以从 subvolumes 的内容可以看出来。

那么这里解析 volfile 文件的代码又是在哪里呢？这是在 libglusterfs/src/graph.ch 和 graph.y 里面，这里 graph.c 中有一个函数 glusterfs_process_svc_attach_volfp，该函数会调用 glusterfs_graph_construct，这就是和构造 volfile 文件有关，另外这里可能不同版本之间的函数名称会有区别，阅读代码的时候请留意。

在 graph.y 文件中有 token 结构，这里就是和定义 volfile 文件结构有关的，同时里面也有 volume_type, new_volume 和 volume_option 等函数，这些也是相关的一些函数，感兴趣的可以自行详细了解。

还有在 volfile 中，如前面的 volume 有 type 有 protocol/server，这个文件又是在哪里的呢？可以简单找一下。

```
1. [root@gfs03 ~]# ls /usr/lib64/glusterfs/9.2/xlator/protocol/
2. client.so  server.so
```

请注意这里不同的操作系统版本，可能存放的文件位置不同。