目录

[理论基础 2](#_Toc47830545)

[1.1分布式文件系统出现 2](#_Toc47830546)

[1.2 典型代表DFS 2](#_Toc47830547)

[1.3面临的问题 3](#_Toc47830548)

[1.4 GlusterFS概述 3](#_Toc47830549)

[1.5 GlusterFS在企业中的应用 3](#_Toc47830550)

[二、安装部署 4](#_Toc47830551)

[2.1安装前准备 4](#_Toc47830552)

[2.2 安装 4](#_Toc47830553)

[2.2.1修改主机名（hostnamectl set-hostname 你的主机名） 4](#_Toc47830554)

[2.2.2 /etc/hosts文件添加DNS解析（IP 、hostname根据自己的实际清理修改） 5](#_Toc47830555)

[2.2.3 安装epel源 5](#_Toc47830556)

[2.2.4 安装glusterfs源 5](#_Toc47830557)

[2.2.5 时间同步 5](#_Toc47830558)

[2.2.6 启动停止服务 6](#_Toc47830559)

[2.2.7 添加集群 6](#_Toc47830560)

[2.2.8配置前的准备工作 7](#_Toc47830561)

[2.2.9 创建volume以及其他操作 8](#_Toc47830562)

[3.构建企业级别的分布式存储 20](#_Toc47830563)

[3.1 硬件要求 20](#_Toc47830564)

[3.2 系统要求和分区划分 20](#_Toc47830565)

[3.3 网络环境要求 20](#_Toc47830566)

[3.4 服务器摆放 21](#_Toc47830567)

[3.5.2 Glusterfs文件系统优化 23](#_Toc47830568)

[3.5.3优化参数调整方式 23](#_Toc47830569)

[3.6 监控及日常维护 24](#_Toc47830570)

[3.7、Gluster日常维护及故障处理 25](#_Toc47830571)

# 理论基础

## 1.1分布式文件系统出现

计算机通过文件系统管理、存储数据，而现在数据信息爆炸的时代中人们可以获取的数据成指数倍的增长，单纯通过增加硬盘个数来扩展计算机文件系统的存储容量的方式，已经不能满足目前的需求。

分布式文件系统可以有效解决数据的存储和管理难题，将固定于某个地点的某个文件系统，扩展到任意多个地点/多个文件系统，众多的节点组成一个文件系统网络。每个节点可以分布在不同的地点，通过网络进行节点间的通信和数据传输。人们在使用分布式文件系统时，无需关心数据是存储在哪个节点上、或者是从哪个节点从获取的，只需要像使用本地文件系统一样管理和存储文件系统中的数据。

## 1.2 典型代表DFS

NFS (Network File system)即网络文件系统，它允许网络中的计算机之间通过TCP/IP网络共享资源。在NFS的应用中，本地NFS的客户端应用可以透明地读写位于远端NFS服务器上的文件，就像访问本地文件一样。 NFS的优点如下:。

1.节约使用的磁盘空间

客户端经常使用的数据可以集中存放在一台机器上，并使用NES发布，那么网络内部所有计算机可以通过网络访问，不必单独存储.+

2.节约硬件资源

NFS还可以共享软驱，CDROM和ZIP等存储设备,减少整个网络上的可移动设备的数量。

3.用户主目录设定

对于特殊用户，如管理员等，为了管理的需要,可能会经常登录到网络中所有的计算机，若每个客户端均保存这个用户的主目录很繁琐,而且不能保证数据的一致性.实际上，经过NFS服务的设定，然后在客户端指定这个用户的主目录位置，并自动挂载，就可以在任何计算机上使用用户主目录的文件。

## 1.3面临的问题

直接用NES挂载存储，有一定风险，存在单点故障。

某些场景不能满足要求，大量的访问磁盘Io是瓶颈。

## 1.4 GlusterFS概述

GlusterFS是Scale-Out存储解决方案Gluster的核心，它是一个开源的分布式文件系统， 具有强大的横向扩展能力，通过扩展能够支持数PB存储容量和处理数千客户端。GlusterFs借助TCP/IP或InfiniBand RDMA 网络将物理分布的存储资源聚集在一起，使用单-全局命名空间来管理数据。GlusterFs基于可堆叠的用户空间设计，可为各种不同的数据负载提供优异的性能。

GlusterFs支持运行在任何标准IP网络上标准应用程序的标准客户端，用户可以在全局统一的命名空间中使用NFS/CIFS等标准协议来访问应用数据。GlusterFs使得用户可摆脱原有的独立、高成本的封闭存储系统，能够利用普通廉价的存储设备来部署可集中管理、横向扩展、虚拟化的存储池，存储容量可扩展至TB/PB级。

## 1.5 GlusterFS在企业中的应用

理论和实践上分析，GlusterFs目前主要适用于**大文件存储场景**，对于小文件尤其是海量小文件，存储效率和访问性能都表现不佳。海量小文件LOSF问题是工业界和学术界公认的难题，GlusterFs作为通用的分布式文件系统，并没有对小文件作额外的优化措施，性能不好也是可以理解的。

Media

-文档、图片、音频、视频。

Shared storage

-云存储、虚拟化存储、HPC (高性能计算)中

Big data

-日志文件、RFID (射频识别)数据。

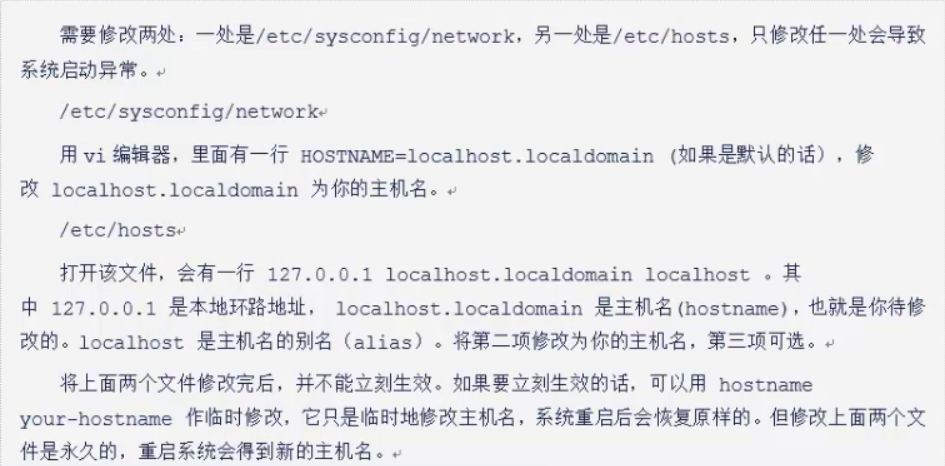
# 二、安装部署

## 2.1安装前准备

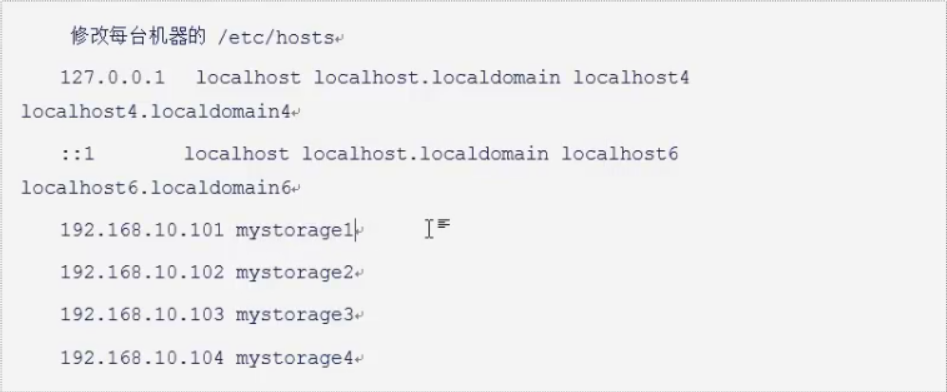
关闭防火墙、selinux、NAT模式、1g运行内存+10G硬盘(第二块硬盘)

## 2.2 安装

### 2.2.1修改主机名（hostnamectl set-hostname 你的主机名）



### 2.2.2 /etc/hosts文件添加DNS解析（IP 、hostname根据自己的实际清理修改）



以下是我的：

[root@node3 ~]# cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.47.191 node1

192.168.47.192 node2

192.168.47.193 node3

192.168.47.194 node4

### 2.2.3 安装epel源

yum –y install epel-release

### 2.2.4 安装glusterfs源

yum -y install centos-release-gluster6.noarch

yum install glusterfs glusterfs-cli glusterfs-fuse glusterfs-server

### 2.2.5 时间同步

[root@node1 ~]# yum -y install ntp ntpdate

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

[root@node3 ~]# hwclock -systohc

[root@node3 ~]# hwclock -w

### 2.2.6 启动停止服务

安装完成后，可以

[root@node1 ~]# glusterfs –V #查看版本号

systemctl start glusterd #启动集群守护进程

systemctl stop glusterd #停止集群守护进程

systemctl status glusterd #查看守护进程状态

systemctl enable glusterd #开机启动守护进程

### 2.2.7 添加集群

**Node1上**

**[root@node1 ~]# gluster peer status**

Number of Peers: 0

[root@node1 ~]# gluster peer probe node2

peer probe: success.

[root@node1 ~]# gluster peer probe node3

peer probe: success.

[root@node1 ~]# gluster peer probe node4

peer probe: success.

#再次查看状态（在每个机器都可以查看）

[root@node1 ~]# gluster peer status

Number of Peers: 3

Hostname: node2

Uuid: 2fa8f1a4-eab3-4531-aeae-f5e96611f472

State: Peer in Cluster (Connected)

Hostname: node3

Uuid: fe895ccd-5651-43eb-8049-61c7e52d45ee

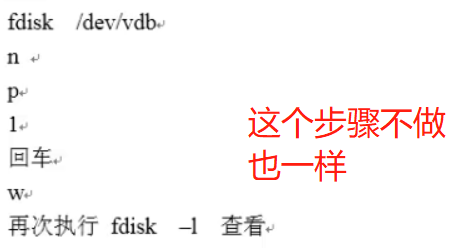
State: Peer in Cluster (Connected)

Hostname: node4

Uuid: e16c79d4-a5b8-4d00-af2f-9a8de235f9b4

State: Peer in Cluster (Connected)

### 2.2.8配置前的准备工作



1. 查看磁盘

[root@node1 ~]# fdisk -l （部分信息）

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors

Disk /dev/sdb: 10.7 GB, 10737418240 bytes, 20971520 sectors

1. 格式化磁盘

官方的格盘方式：<https://docs.gluster.org/en/latest/Quick-Start-Guide/Quickstart/#purpose-of-this-document>

教程中的：（4台均操作）

[root@node1 ~]# mkfs.xfs -f /dev/sdb

新建节点+挂载：（4台均操作）

[root@node1 ~]# mkdir -p /storage/brick1

#挂载

[root@node1 ~]# mount /dev/sdb /storage/brick1/

#查看挂载情况

[root@node3 ~]# df -h

Filesystem Size Used Avail Use% Mounted on

/dev/mapper/centos-root 17G 1.2G 16G 7% /

devtmpfs 475M 0 475M 0% /dev

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

tmpfs 487M 7.6M 479M 2% /run

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

/dev/sda1 1014M 133M 882M 14% /boot

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

/dev/sdb 10G 33M 10G 1% /storage/brick1

#自动挂载

echo '/dev/sdb / storage /brick1 xfs defaults 0 0' >> /etc/fstab

#挂载  
mount -a

df -h

emmm：多块盘就格式多次（例如mkfs.xfs -f /dev/sdc、mkfs.xfs -f /dev/sdd），然后目录就新建多个（mkdir -p /storage/brick2，mkdir -p /storage/brick3）多块自动挂载{

echo '/dev/sdb / storage /brick1 xfs defaults 0 0' >> /etc/fstab

echo '/dev/sdc / storage /brick1 xfs defaults 0 0' >> /etc/fstab

}

### 2.2.9 创建volume以及其他操作

GlusterFS 五种卷

* Distributed：分布式卷，文件通过 hash 算法随机分布到由 bricks 组成的卷上。
* Replicated: 复制式卷，类似 RAID 1，replica 数必须等于 volume 中 brick 所包含的存储服务器数，可用性高。
* Striped: 条带式卷，类似 RAID 0，stripe 数必须等于 volume 中 brick 所包含的存储服务器数，文件被分成数据块，以 Round Robin 的方式存储在 bricks 中，并发粒度是数据块，大文件性能好。
* Distributed Striped: 分布式的条带卷，volume中 brick 所包含的存储服务器数必须是 stripe 的倍数（>=2倍），兼顾分布式和条带式的功能。
* Distributed Replicated: 分布式的复制卷，volume 中 brick 所包含的存储服务器数必须是 replica 的倍数（>=2倍），兼顾分布式和复制式的功能。

　分布式复制卷的brick顺序决定了文件分布的位置，一般来说，先是两个brick形成一个复制关系，然后两个复制关系形成分布。

　企业一般用后两种，大部分会用分布式复制（可用容量为 总容量/复制份数），通过网络传输的话最好用万兆交换机，万兆网卡来做。这样就会优化一部分性能。它们的数据都是通过网络来传输的。

#创建分布式卷

[root@node1 ~]# gluster volume create gv1 node1:/storage/brick1 node2:/storage/brick1 force

volume create: gv1: success: please start the volume to access data

语法：（我们现在做的是node1和node2上的卷，在3、4也是可以看到的gluster volume info）

gv1：名字自己命名

node1：主机名

/storage/brick1：刚刚我们新建的brick1节点

[root@node1 ~]# gluster volume info

Volume Name: gv1

Type: Distribute（分布式）

Volume ID: b6ccfa7b-ee61-419e-9e53-ab320da73a92

Status: Created

Snapshot Count: 0

Number of Bricks: 2

Transport-type: tcp（连接方式）

Bricks:

Brick1: node1:/storage/brick1

Brick2: node2:/storage/brick1

Options Reconfigured:

transport.address-family: inet nfs.disable: on

#启动卷

[root@node1 ~]# gluster volume start gv1

volume start: gv1: success

[root@node1 ~]# gluster volume stop VOLNAME #停止卷

[root@node1 ~]# gluster volume delete VOLNAME #删除卷

#挂载卷到目录

Node1上

[root@node1 ~]# mount -t glusterfs 127.0.0.1:/gv1 /mnt/

[root@node1 ~]# df –h #查看挂载情况（大小是20g是因为10+10 也就是2台机器的和）

Filesystem Size Used Avail Use% Mounted on

127.0.0.1:/gv1 20G 270M 20G 2% /mnt

Node3上

[root@node3 ~]# mount -t glusterfs 127.0.0.1:/gv1 /mnt/

[root@node3 ~]# df –h

Filesystem Size Used Avail Use% Mounted on

127.0.0.1:/gv1 20G 270M 20G 2% /mnt

Node3上

[root@node3 mnt]# cd /mnt/

[root@node3 mnt]# touch a b c d f

[root@node3 mnt]# ls

a b c d f

Node1上（也可以看到刚刚新建的几个文件）【事实上任意节点挂载卷到目录都可以看到这几个文件】

[root@node1 mnt]# cd /mnt/

[root@node1 mnt]# ls

a b c d f

#用NFS方式挂载

#先安装nfs

[root@node4 ~]# yum install nfs-utils

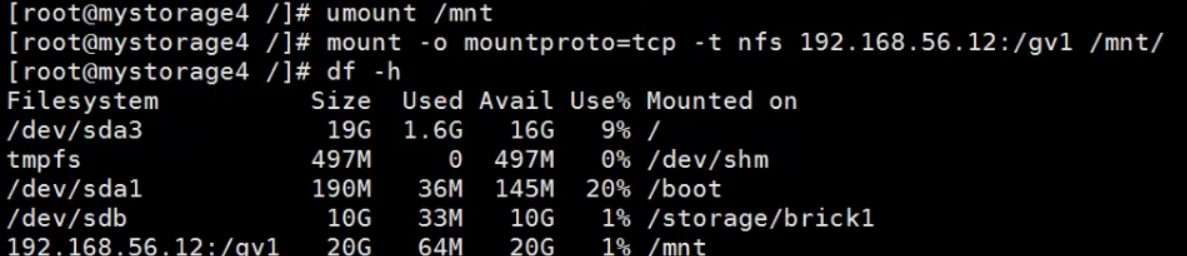
[root@node4 ~]# mount -o mountproto=tcp -t nfs node1:/gv1 /mnt/

mount.nfs: requested NFS version or transport protocol is not supported

报错参考：（没时间弄）

<https://www.cnblogs.com/zihanxing/articles/6004202.html>

正确结果如下：



#创建复制卷

#可以理解成跨主机的raid1（数据分相同的2份）

#创建命令

[root@node1 ~]# gluster volume create gv2 replica 2 node3:/storage/brick1 node4:/storage/brick1 force

结果：volume create: gv2: success: please start the volume to access data

#参数解释

gv2：自定义的名字

replica 2：分两份，所有后面写2台主机：/目录（如果分3份就写三台的）

node3:/storage/brick1：主机：/目录（目录在之前创建好了）

#启动（任意节点启动都可以）

[root@node2 ~]# gluster volume start gv2

volume start: gv2: success

#挂载（挂载目录可以自己选的）

[root@node1 ~]# mount -t glusterfs 127.0.0.1:/gv2 /opt/

[root@node1 ~]# df -h

[root@node3 ~]# df –h（10G类似raid1）#容量是总容量的一半

Filesystem Size Used Avail Use% Mounted on

127.0.0.1:/gv1 20G 270M 20G 2% /mnt

127.0.0.1:/gv2 10G 135M 9 .9G 2% /opt

#测试 node1上

[root@node1 ~]# cd /opt/

[root@node1 opt]# ll

total 0

[root@node1 opt]# touch t z h

[root@node1 opt]# ls

h t z

#查看结果 node3上（node4也可）

[root@node3 ~]# cd /storage/brick1/

[root@node3 brick1]# ls

h t z

#node2上是没有的

[root@node2 ~]# cd /storage/brick1/

[root@node2 brick1]# ls

d f

#查看信息

[root@node2 ~]# gluster volume info gv2

Volume Name: gv2

Type: Replicate #复制卷

Volume ID: c99aefd9-9b22-47b1-8c9e-9249a6573458

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: node3:/storage/brick1

Brick2: node4:/storage/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

**文件实际存在位置node3和node4上的/storage/brick1目录下,通过hash分别存到node01和node02上的分布式磁盘上**

#配置分布式复制卷（前面用node3、4现在用node1、2）

最少需要4台服务器才能创建,[生产场景推荐使用此种方式]

#将原有的复制卷gv2进行扩容，使其成为分布式复制卷；

#要扩容前需停掉gv2

 [root@node01 ~]# gluster volume stop gv2

#添加brick到gv2中（前面用brick1，现在用brick2）

[root@node1 ~]# gluster volume add-brick gv2 replica 2 node1:/storage/brick2 node2:/storage/brick2 force

volume add-brick: success

#启动

[root@node01 ~]# gluster volume start gv2

volume start: gv2: success

#查看信息

[root@node1 ~]# gluster volume info gv2

Volume Name: gv2

Type: Distributed-Replicate

Volume ID: a451fa10-2dda-45d8-8fc0-fcb7f8da0d82

Status: Started

Snapshot Count: 0

Number of Bricks: 2 x 2 = 4

Transport-type: tcp

Bricks:

Brick1: node3:/storage/brick1

Brick2: node4:/storage/brick1

Brick3: node1:/storage/brick2

Brick4: node2:/storage/brick2

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

注意：

当你给分布式复制卷和分布式条带卷增加 bricks 时，你增加的 bricks 数目必须是复制或条带数目的倍数，例如：你给一个分布式复制卷的 replica 为 2，你在增加 bricks 的时候数量必须为2、4、6、8等。扩容后进行测试，发现文件都分布在扩容前的卷中。

#创建条带卷

#保留之前做的实验，所以在每台机器上新增一块5G的硬盘（加完reboot）

#如果不想添加可以把之前挂载的/mnt /opt卸载（[root@node1 ~]# umount /mnt/ /opt/）

#不过练习为主还是再加一块吧！

#四台机器都操作啊！！！（重启之后刚刚的实验也没得了因为没写配置文件到/etc/fstab 尴尬）

1. 通过fdisk –l 查看刚刚添加的硬盘信息

[root@node1 ~]# fdisk –l

Disk /dev/sdc: 5368 MB, 5368709120 bytes, 10485760 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

1. 格式化

[root@node1 ~]# mkfs.xfs -f /dev/sdc

1. 新建目录

[root@node1 ~]# mkdir -p /storage/brick2

1. 自动挂载

[root@node1 ~]# echo '/dev/sdc /storage/brick2 xfs defaults 0 0' >> /etc/fstab

[root@node1 ~]# mount -a

1. 查看挂载

[root@node1 ~]# df –h

/dev/sdc 5.0G 33M 5.0G 1% /storage/brick2

#正式创建

[root@node1 ~]# gluster volume create gv3 stripe 2 node1:/storage/brick2 node1:/storage/brick2 force

stripe option not supported（啊这这查资料说不支持了）

#启动（启动个球球哦不支持了）

[root@node1 ~]# gluster volume start gv3

#查看信息

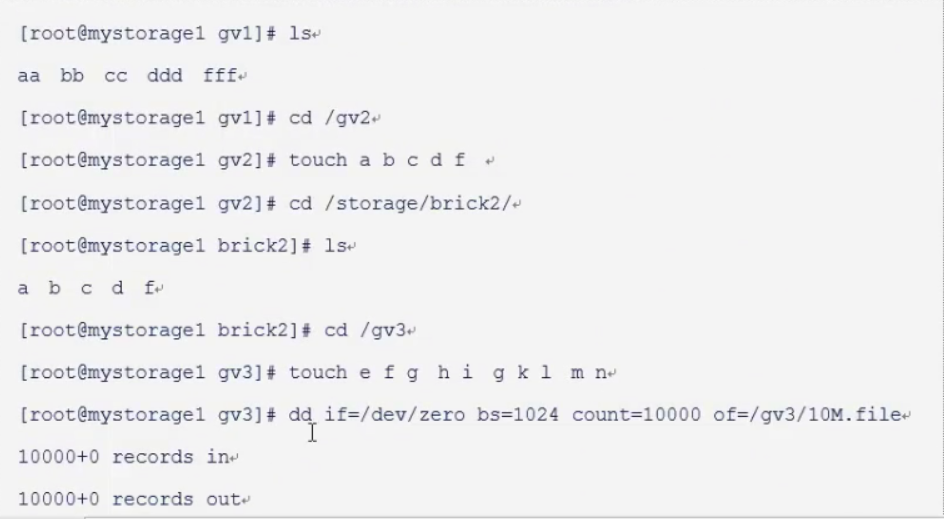


#挂载命令：（正式来说挂载的目录自己建立）

[root@node1 ~]# mkdir /gv3

[root@node1 ~]# mount -t glusterfs 127.0.0.1:/gv3 / gv3

#测试看图吧



#配置分布式条带卷（前面用node1、2现在用node3、4）

#将原有的复制卷gv3进行扩容，使其成为分布式条带卷

#要扩容前需停掉gv3

[root@node01 ~]# gluster volume stop gv3

#添加brick到gv3中

[root@node01 ~]# gluster volume add-brick gv3 stripe 2 node03:/storage/brick2 node04:/ storage/brick2 force

#启动gv3

[root@node01 ~]# gluster volume start gv3

volume start: gv3: success

#查看详细信息

[root@node01 ~]# gluster volume info gv3

Volume Name: gv3

Type: Distributed-Stripe # 这里显示是分布式条带卷，是在 gv3 条带卷的基础上增加 2 块 brick 形成的

Volume ID: 54c16832-6bdf-42e2-81a9-6b8d7b547c1a

Status: Started

Snapshot Count: 0

Number of Bricks: 2 x 2 = 4

Transport-type: tcp

Bricks:

Brick1: node01:/ storage /brick3

Brick2: node02:/ storage /brick3

Brick3: node03:/ storage /brick2

Brick4: node04:/ storage /brick2

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

分布式复制卷的最佳实践：

　1)搭建条件  
　 - 块服务器的数量必须是复制的倍数  
 - 将按块服务器的排列顺序指定相邻的块服务器成为彼此的复制  
例如，8台服务器：  
 - 当复制副本为2时，按照服务器列表的顺序，服务器1和2作为一个复制,3和4作为一个复制,5和6作为一个复制,7和8作为一个复制  
 - 当复制副本为4时，按照服务器列表的顺序，服务器1/2/3/4作为一个复制,5/6/7/8作为一个复制

磁盘存储的平衡  
平衡布局是很有必要的，因为布局结构是静态的，当新的 bricks 加入现有卷，新创建的文件会分布到旧的 bricks 中，所以需要平衡布局

#在gv2的分布式复制卷的挂载目录中创建测试文件入下

[root@node01 ~]# df -h

文件系统 容量 已用 可用 已用% 挂载点

127.0.0.1:/gv2 15G 219M 15G 2% /opt

#新创建的文件只在老的brick中有，在新加入的brick中是没有的

[root@node01 ~]# cd /opt/

[root@node1 opt]# touch test

[root@node4 ~]# ls /storage/brick1/（我这里node3、4是老的）

fsfs test y

[root@node3 ~]# ls /storage/brick1/（我这里node3、4是老的）

fsfs test y

[root@node1 opt]# ll -h /storage/brick2

total 0

[root@node2 ~]# ll -h /storage/brick2

total 0

# 从上面可以看到，新创建的文件还是在之前的 bricks 中，并没有分布中新加的 bricks 中

# 下面进行磁盘存储平衡

[root@node1 ~]# gluster volume rebalance gv2 start

volume rebalance: gv2: success: Rebalance on gv2 has been started successfully. Use rebalance status command to check status of the rebalance process.

ID: f9faadc8-6315-4771-a411-8f1cedd48e40

[root@node1 ~]# gluster volume rebalance gv2 status

#查看平衡存储状态

# 查看磁盘存储平衡后文件在 bricks 中的分布情况

[root@node3 ~]# ls /storage/brick1

[root@node4 ~]# ls /storage/brick1

[root@node1 ~]# ls /storage/brick2

[root@node2 ~]# ls /storage/brick2

#从上面可以看出**部分文件**已经平衡到新加入的brick中了每做一次扩容后都需要做一次磁盘平衡。 磁盘平衡是在万不得已的情况下再做的，一般再创建一个卷就可以了。

删除卷  
一般会用在命名不规范的时候才会删除

[root@node1 ~]# gluster volume stop gv1

[root@node1 ~]# gluster volume delete gv1

#移除brick

你可能想在线缩小卷的大小，例如：当硬件损坏或网络故障的时候，你可能想在卷中移除相关的 bricks。  
　　注意：当你移除 bricks 的时候，你在 gluster 的挂载点将不能继续访问数据，只有配置文件中的信息移除后你才能继续访问 bricks 中的数据。当移除分布式复制卷或者分布式条带卷的时候，移除的 bricks 数目必须是 replica 或者 stripe 的倍数。

　　但是移除brick在生产环境中基本上不做的，如果是硬盘坏掉的话，直接换个好的硬盘即可，然后再对新的硬盘设置卷标识就可以使用了，后面会演示硬件故障或系统故障的解决办法。

#移除步骤

#先停止

[root@node1 ~]# gluster volume stop gv2

[root@node1 ~]# gluster volume remove-brick gv2 replica 2 node3:/storage/brick1 node4:/storage/brick1 force

volume remove-brick commit force: success

[root@node1 ~]# gluster volume info gv2

Volume Name: gv2

Type: Replicate

Volume ID: a451fa10-2dda-45d8-8fc0-fcb7f8da0d82

Status: Stopped

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: node1:/storage/brick2

Brick2: node2:/storage/brick2

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

# 如果误操作删除了后，其实文件还在 /storage/brick1 里面的，加回来就可以了

[root@node1 ~]# gluster volume add-brick gv2 replica 2 node3:/storage/brick1 node4:/storage/brick1 force

volume add-brick: success

[root@node1 ~]# gluster volume info gv2

Volume Name: gv2

Type: Distributed-Replicate

Volume ID: a451fa10-2dda-45d8-8fc0-fcb7f8da0d82

Status: Stopped

Snapshot Count: 0

Number of Bricks: 2 x 2 = 4

Transport-type: tcp

Bricks:

Brick1: node1:/storage/brick2

Brick2: node2:/storage/brick2

Brick3: node3:/storage/brick1

Brick4: node4:/storage/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

#模拟误删除卷信息故障及解决办法

#查看卷

[root@node1 ~]# ls /var/lib/glusterd/vols/

gv1 gv2

#干掉其中一个

[root@node1 ~]# rm -rf /var/lib/glusterd/vols/gv1/

#恢复

[root@node1 ~]# gluster volume sync node2

Sync volume may make data inaccessible while the sync is in progress. Do you want to continue? (y/n) y

[root@node01 ~]# ls /var/lib/glusterd/vols/ #验证卷信息是否同步过来

[root@node1 vols]# scp -r node2:/var/lib/glusterd/vols/gv1/ . #？试试这个

#模拟复制卷数据不一致故障及解决办法

#查看/storage/brick2

[root@node1 ~]# ls /storage/brick2

sds sdsd x

#干掉

[root@node1 ~]# rm -rf /storage/brick2/\*

#查看节点node2的

[root@node2 ~]# ls /storage/brick2

sds sdsd x z

#因为之前关闭了，如果未关闭可以忽略此步

[root@node1 ~]# gluster start gv2

#通过访问这个复制卷的挂载点的数据来同步数据

[root@node1 ~]# cat /opt

#这时候再看复制卷的数据是否同步成功

[root@node1 opt]# ls /storage/brick2/

sds sdsd x z

# 3.构建企业级别的分布式存储

### 3.1 硬件要求

一般选择2U的机型，磁盘STAT 盘4T，如果I/O要求比较高，可以采购SSD固态硬盘。

为了充分保证系统的稳定性和性能,要求所有glusterfs 服务器硬件配置尽量一致， 尤其是硬盘数量和大小。机器的RAID卡需要带电池,缓存越大,性能越好。一般情况下,建议做RAID10,如果出于空间要求的考虑，需要做RAID5，建议最好能有1-2块硬盘的热备盘。

### 3.2 系统要求和分区划分

系统要求使用centos 6.x,安装完成后升级到最新版本，安装的时候,不要使用LV,建议/boot分区200M,/分区100G、swap 分区和内存一样大小， 剩余空间给gluster使用，划分单独的硬盘空间。系统安装软件没有特殊要求，建议除了开发工具和基本的管理软件，其他软件一律不安装。

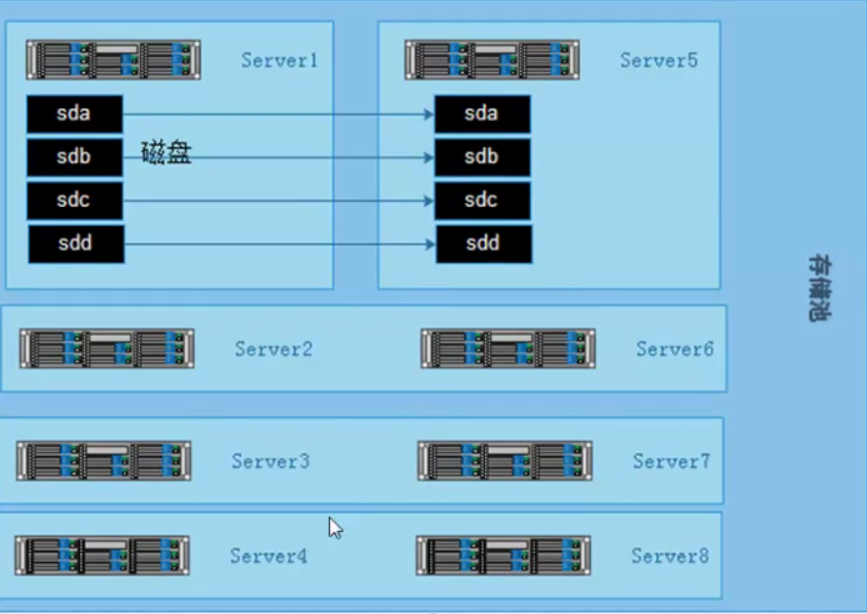
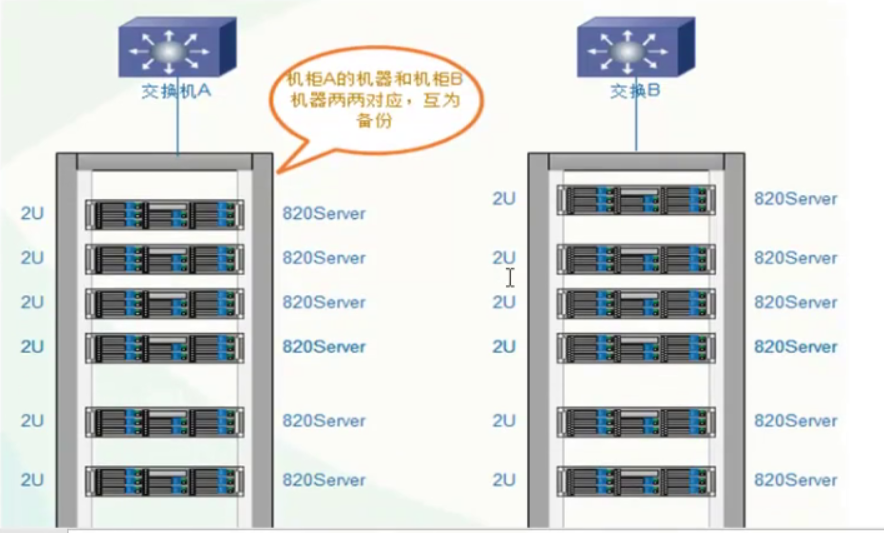
### 3.3 网络环境要求

网络要求全部千兆环境，gluster 服务器至少有2块网卡，1块网卡绑定供gluster使用，剩余一块分配管理网络ip,用于系统管理。如果有条件购买万兆交换机,服务器配置万兆网卡，

存储性能性能会更好。网络方面如果安全性要求较高，可以多网卡绑定。

### 3.4 服务器摆放

服务器主备机器要放在不同的机柜，连接不同的交换机， 即使一个机构出现问题，还有一份数据正常访问。





[root@node1 ~]# cat /etc/glusterfs/glusterd.vol

volume management

type mgmt/glusterd

option working-directory /var/lib/glusterd

option transport-type socket,rdma

option transport.socket.keepalive-time 10

option transport.socket.keepalive-interval 2

option transport.socket.read-fail-log off

option transport.socket.listen-port 24007

option transport.rdma.listen-port 24008

option ping-timeout 0

option event-threads 1

# option lock-timer 180

# option transport.address-family inet6

# option base-port 49152

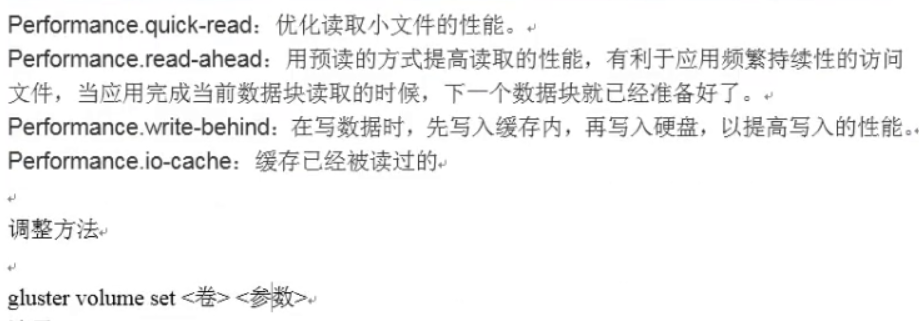
option max-port 60999

end-volume

### 3.5.2 Glusterfs文件系统优化



### 3.5.3优化参数调整方式



命令格式：

gluster.volume set <卷><参数>

#查看原本的

[root@node1 ~]# gluster volume info gv2

Volume Name: gv2

Type: Distributed-Replicate

Volume ID: a451fa10-2dda-45d8-8fc0-fcb7f8da0d82

Status: Started

Snapshot Count: 0

Number of Bricks: 2 x 2 = 4

Transport-type: tcp

Bricks:

Brick1: node1:/storage/brick2

Brick2: node2:/storage/brick2

Brick3: node3:/storage/brick1

Brick4: node4:/storage/brick1

Options Reconfigured:

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

#打开预读方式访问存储

[root@node1 ~]# gluster volume set gv2 performance.read-ahead on

volume set: success

#调整读取缓存的大小  
[root@node1 ~]# gluster volume set gv2 performance.cache-size 256MB

volume set: success

#再次查看

[root@node1 ~]# gluster volume info gv2

Volume Name: gv2

Type: Distributed-Replicate

Volume ID: a451fa10-2dda-45d8-8fc0-fcb7f8da0d82

Status: Started

Snapshot Count: 0

Number of Bricks: 2 x 2 = 4

Transport-type: tcp

Bricks:

Brick1: node1:/storage/brick2

Brick2: node2:/storage/brick2

Brick3: node3:/storage/brick1

Brick4: node4:/storage/brick1

Options Reconfigured:

performance.cache-size: 256MB

performance.read-ahead: on

transport.address-family: inet

nfs.disable: on

### 3.6 监控及日常维护

使用zabbix自带的模板即可，CPU、内存、磁盘空间、主机运行时间、系统load。日常情况要查看服务器监控值，遇到报警要及时处理。

#看下节点有没有在线

gluster volume status gv2

#启动完全修复

gluster volume heal gv2 full

#查看需要修复的文件

gluster volume heal gv2 info

#查看修复成功的文件

gluster volume heal gv2 info healed

#查看修复失败的文件

gluster volume heal gv2 heal-failed

#查看主机的状态

gluster peer status

#查看脑裂的文件

gluster volume heal gv2 info split-brain

#激活quota功能

gluster volume quota gv2 enable

#关闭quota功能

gulster volume quota gv2 disable

#目录限制（卷中文件夹的大小）

gluster volume quota limit-usage /data/30MB --/gv2/data

#quota信息列表

gluster volume quota gv2 list

#限制目录的quota信息

gluster volume quota gv2 list /data

#设置信息的超时时间

gluster volume set gv2 features.quota-timeout 5

#删除某个目录的quota设置

gluster volume quota gv2 remove /data

备注：quota功能，主要是对挂载点下的某个目录进行空间限额。如：/mnt/gulster/data目录，而不是对组成卷组的空间进行限制。

### 3.7、Gluster日常维护及故障处理

1、硬盘故障

如果底层做了raid配置，有硬件故障，直接更换硬盘，会自动同步数据。

如果没有做raid处理方法：  
参看其他博客：<http://blog.51cto.com/cmdschool/1908647>

2、一台主机故障

<https://www.cnblogs.com/xiexiaohua007/p/6602315.html>

一台节点故障的情况包含以下情况：  
  
物理故障:  
同时有多块硬盘故障，造成数据丢失  
系统损坏不可修复  
解决方法：  
  
找一台完全一样的机器，至少要保证硬盘数量和大小一致，安装系统，配置和故障机同样的 IP，安装 gluster 软件，  
保证配置一样，在其他健康节点上执行命令 gluster peer status，查看故障服务器的 uuid

[root@mystorage2 ~]# gluster peer status

Number of Peers: 3

Hostname: mystorage3

Uuid: 36e4c45c-466f-47b0-b829-dcd4a69ca2e7

State: Peer in Cluster (Connected)

Hostname: mystorage4

Uuid: c607f6c2-bdcb-4768-bc82-4bc2243b1b7a

State: Peer in Cluster (Connected)

Hostname: mystorage1

Uuid: 6e6a84af-ac7a-44eb-85c9-50f1f46acef1

State: Peer in Cluster (Disconnected)

复制代码

修改新加机器的 /var/lib/glusterd/glusterd.info 和 故障机器一样

[root@mystorage1 ~]# cat /var/lib/glusterd/glusterd.info

UUID=6e6a84af-ac7a-44eb-85c9-50f1f46acef1

operating-version=30712

在信任存储池中任意节点执行

# gluster volume heal gv2 full

就会自动开始同步，但在同步的时候会影响整个系统的性能。

可以查看状态

# gluster volume heal gv2 info