

GlusterFS是什么？



1、GlusterFS简介

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，具有强大的横向拓展能力，通过拓展能够支持数PB存储容量和数千客户端。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，使用单一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。具有可拓展性、高性能、高可用性、全局统一命名空间、弹性哈希算法弹性卷管理、基于标准协议等特点。

2、GlusterFS架构

GlusterFS 架构 特 点

软件定义

无中心架构

全局命名空间

高性能

用户空间实现

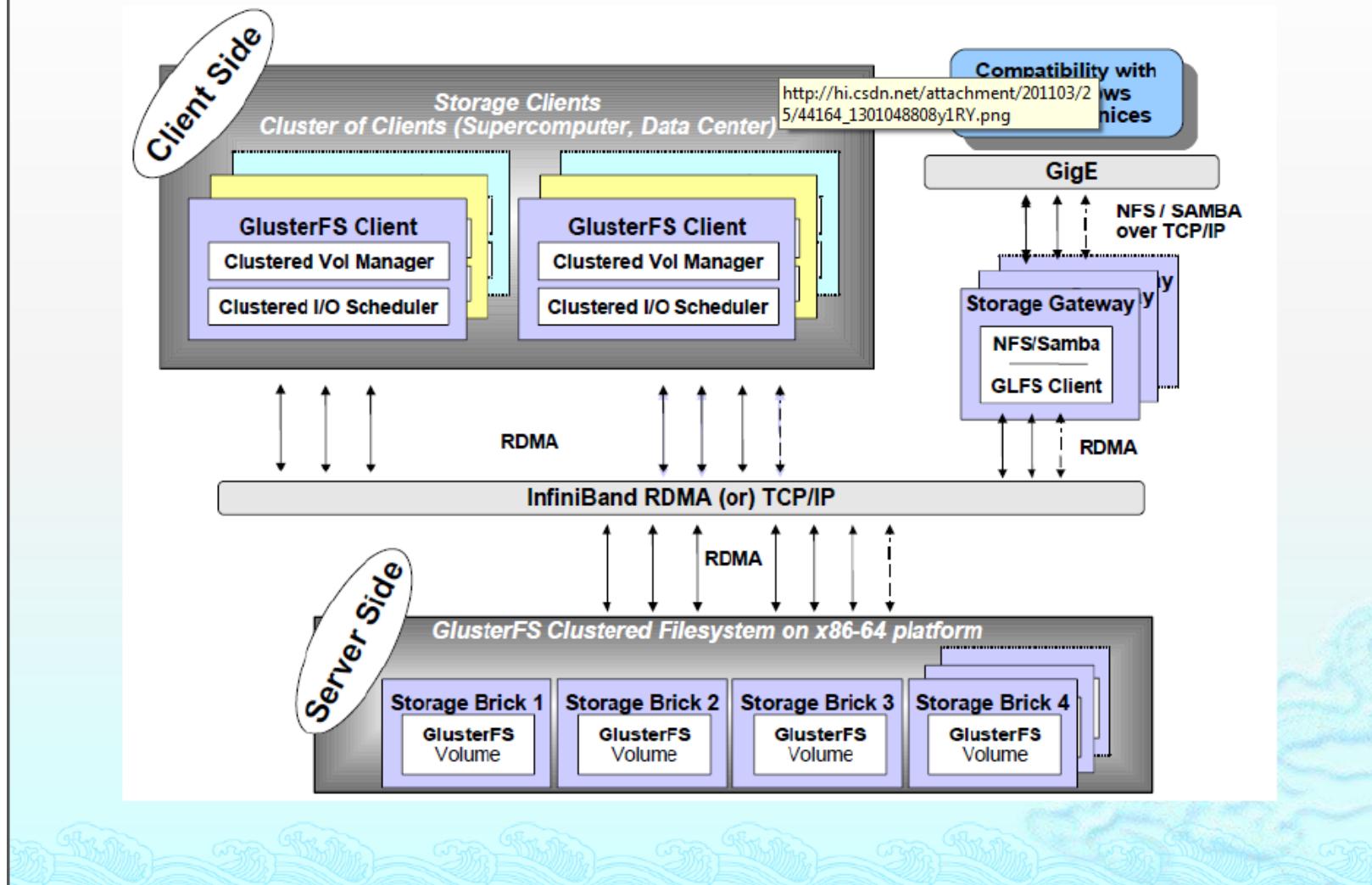
堆栈式设计

弹性横向扩展

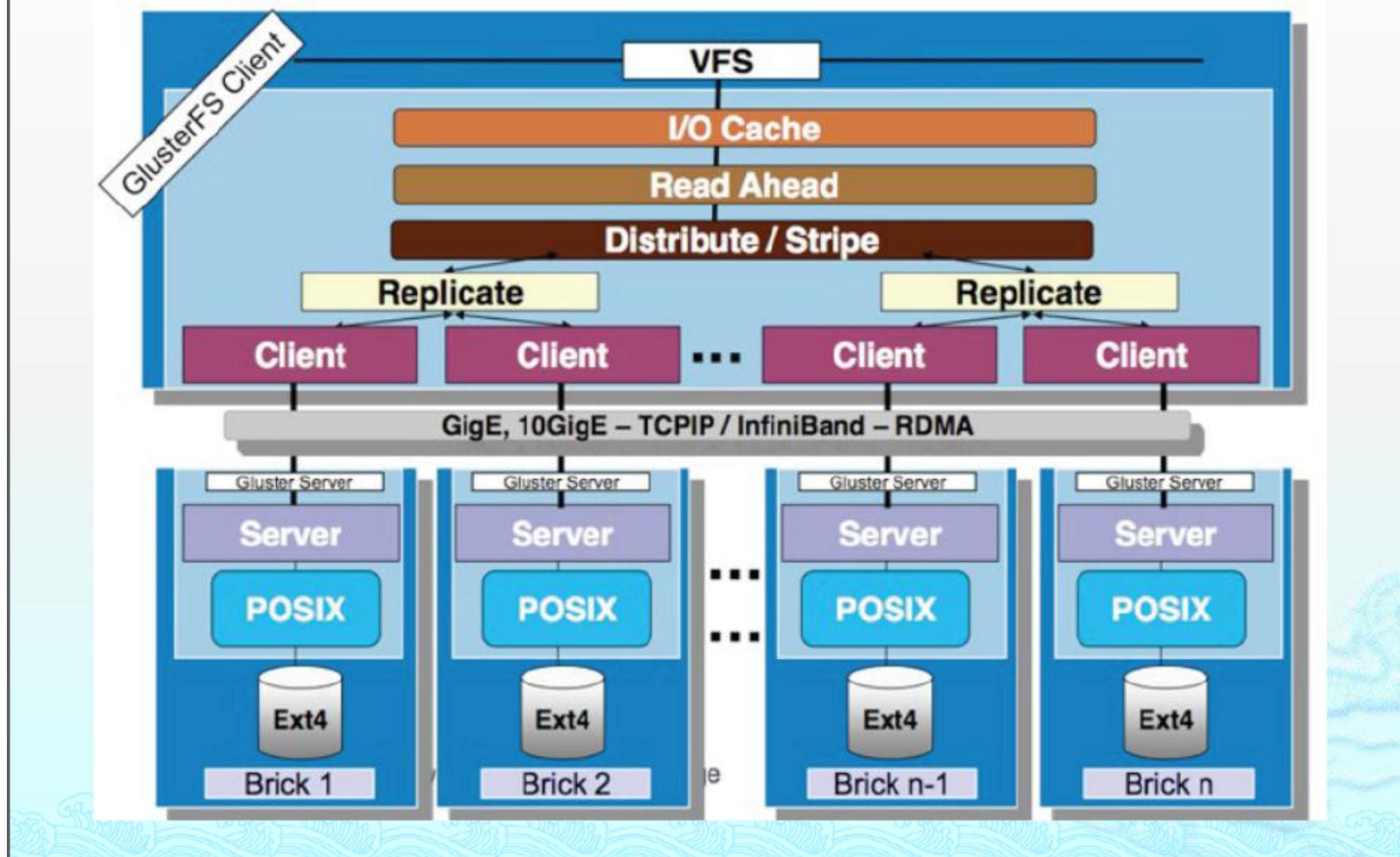
高速网络通信

数据自动修复

GlusterFS 总体架构

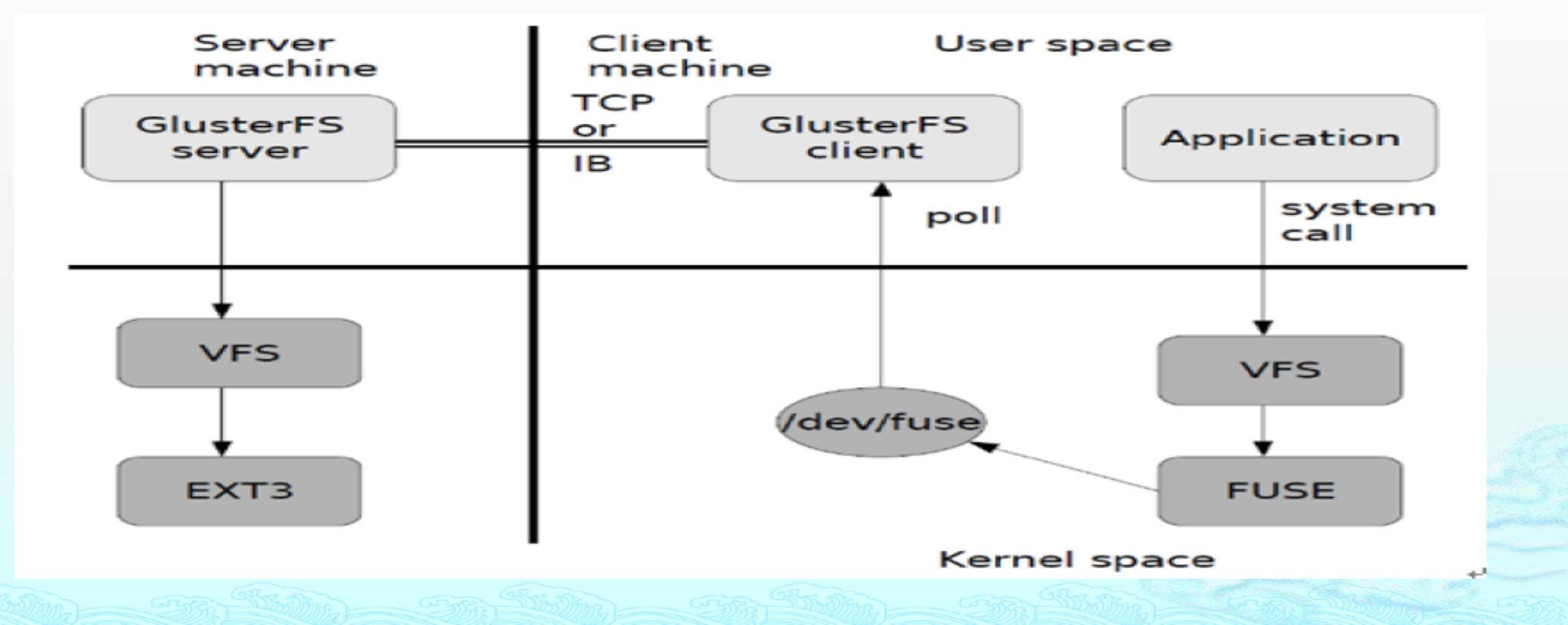


GlusterFS 堆栈式软件架构



3、GlusterFS数据流

GlusterFS 数据 流



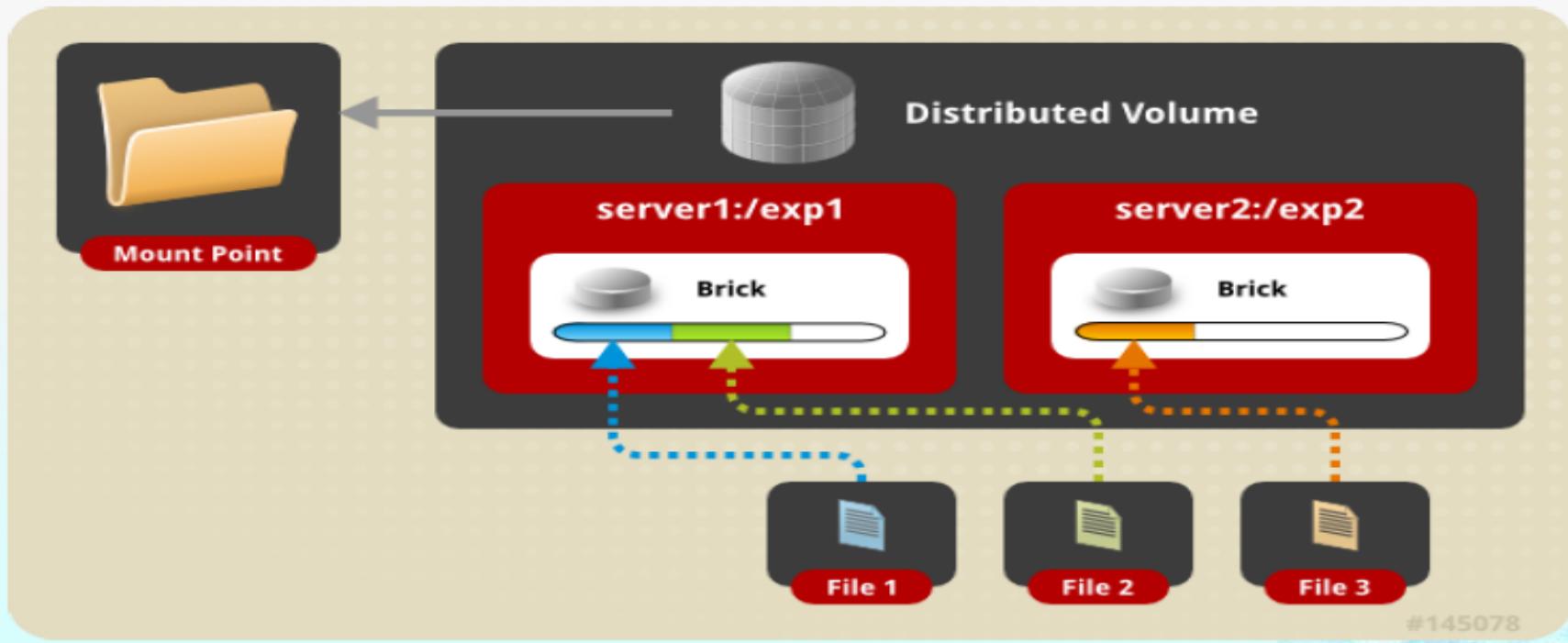
4、GlusterFS卷类型

GlusterFS 卷 类 型

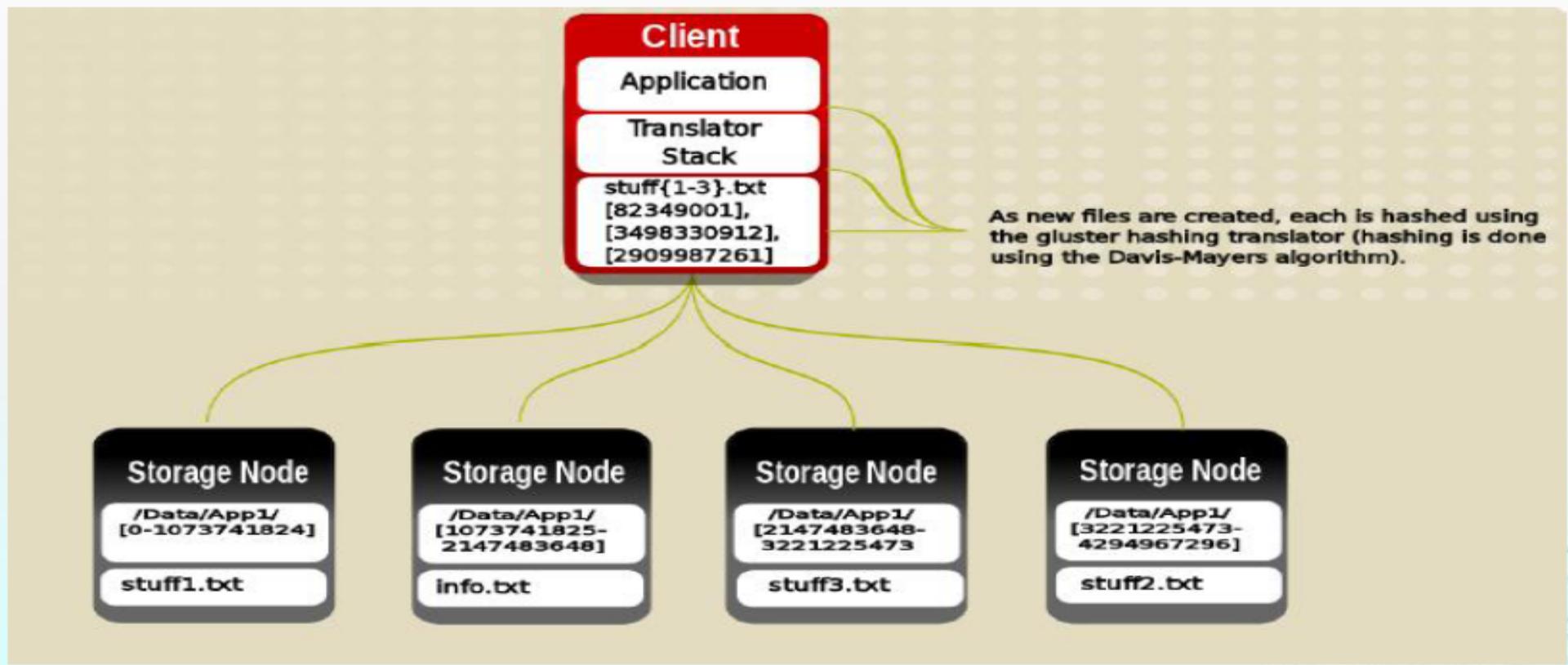
- ◆ 基本卷
 - ◆ 哈希卷 (Distributed Volume)
 - ◆ 复制卷 (Replicated Volume)
 - ◆ 条带卷 (Striped Volumes)
- ◆ 复合卷
 - ◆ 哈希复制卷(Distributed Replicated Volume)
 - ◆ 哈希条带卷 (Distributed Striped Volume)
 - ◆ 复制条带卷 (Replicated Striped Volume)
 - ◆ 哈希复制条带卷 (Distributed Replicated Striped Volume)

哈希卷 (Distributed Volume)

- 文件通过hash算法在所有brick上分布
- 文件级RAID 0，不具有容错能力

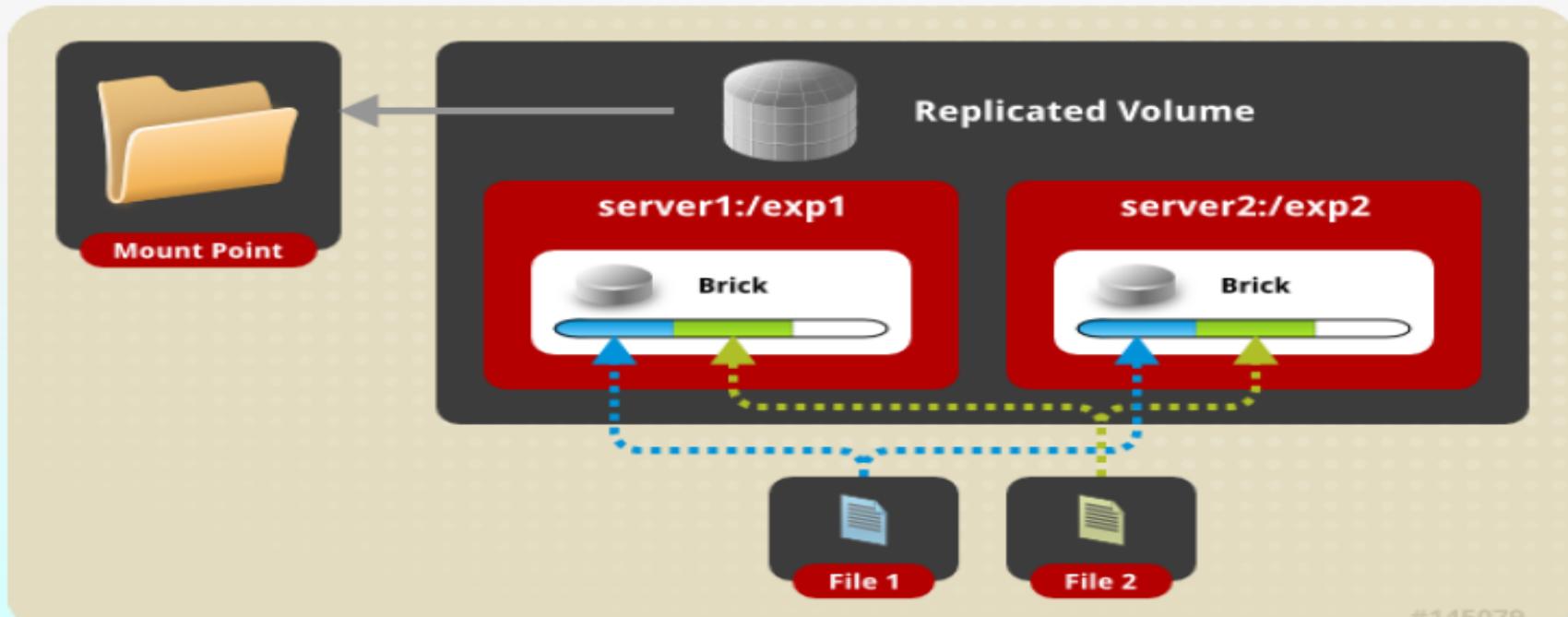


哈希卷工作原理

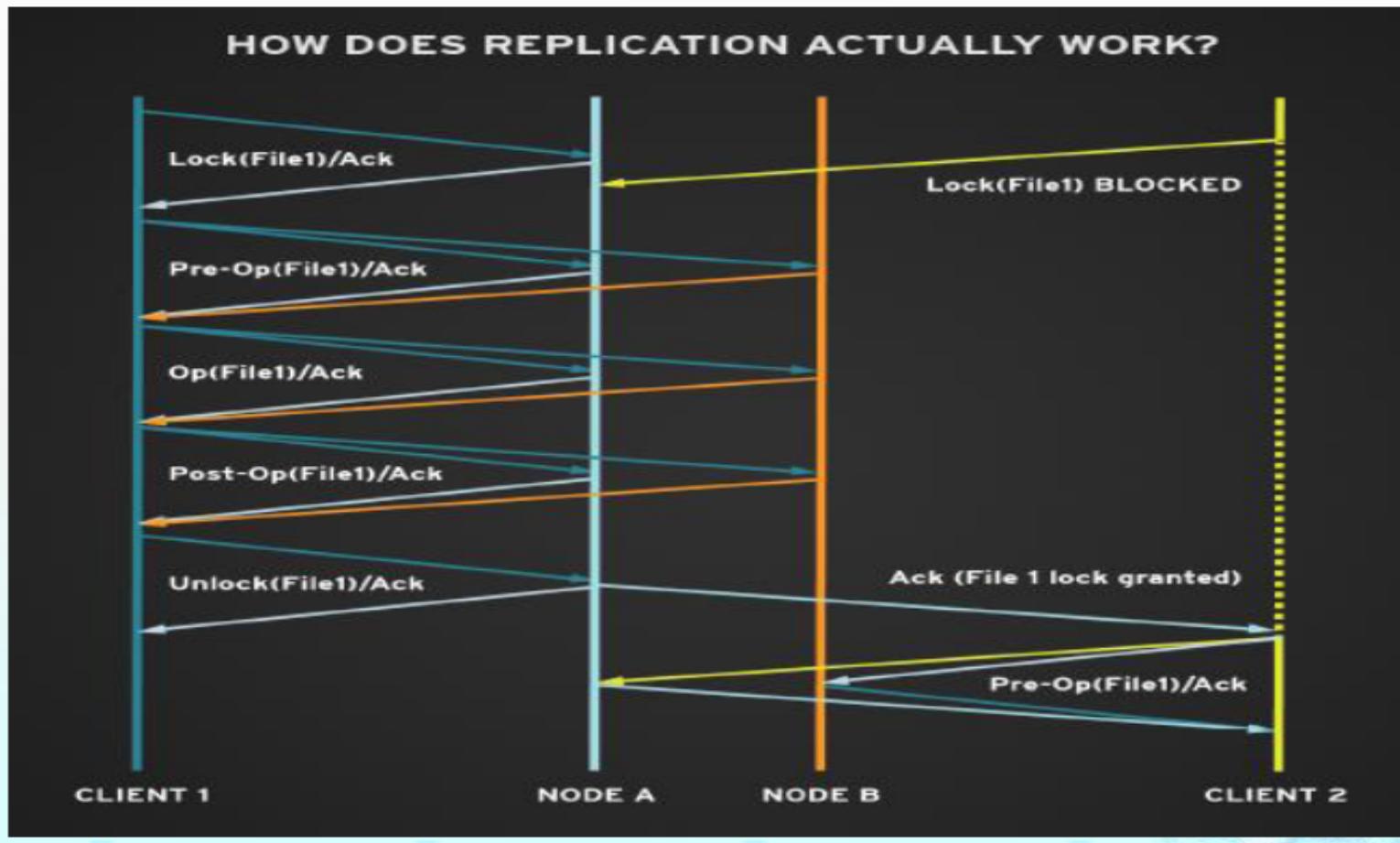


复制卷 (Replicated Volume)

- 文件同步复制到多个brick上
- 文件级RAID 1，具有容错能力
- 写性能下降，读性能提升

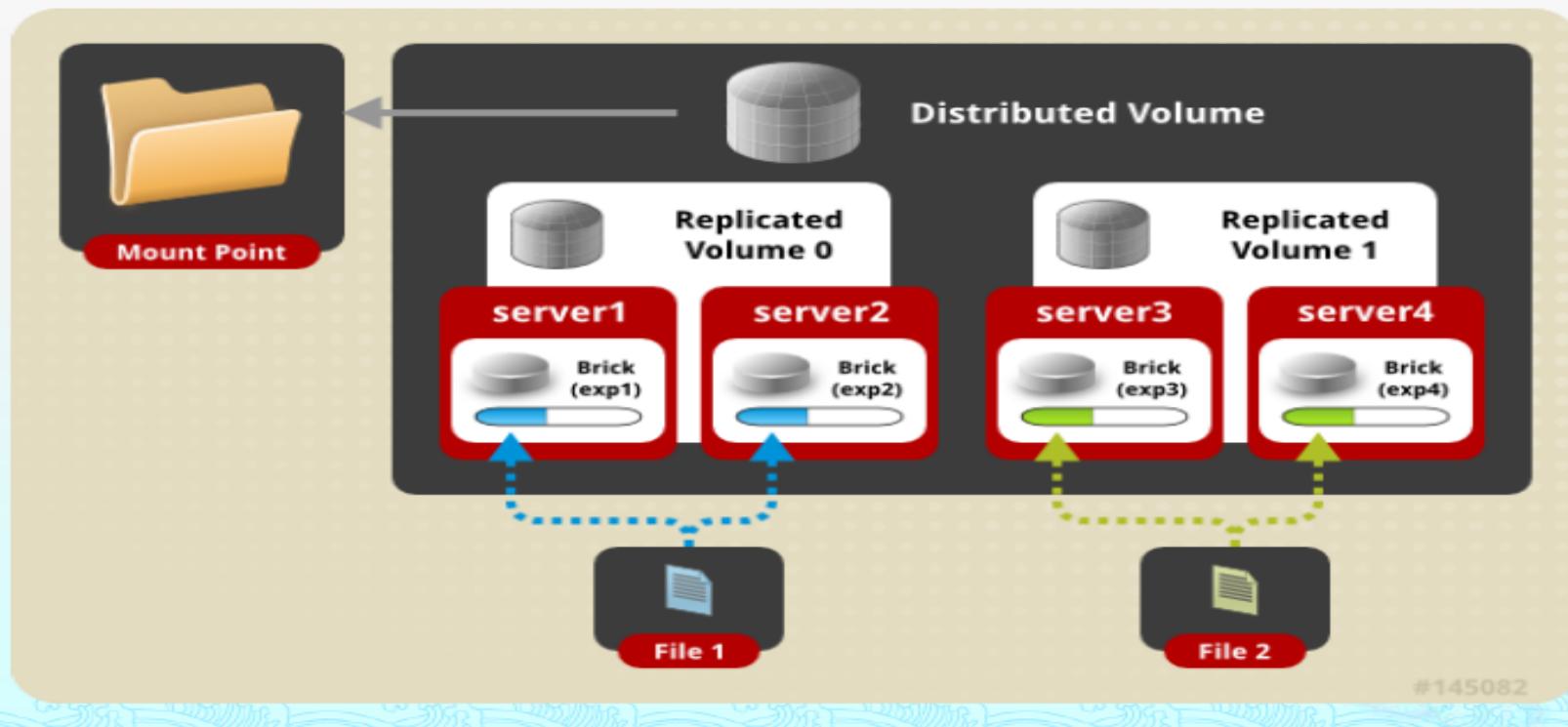


复制卷工作原理



复合卷：哈希 + 复制

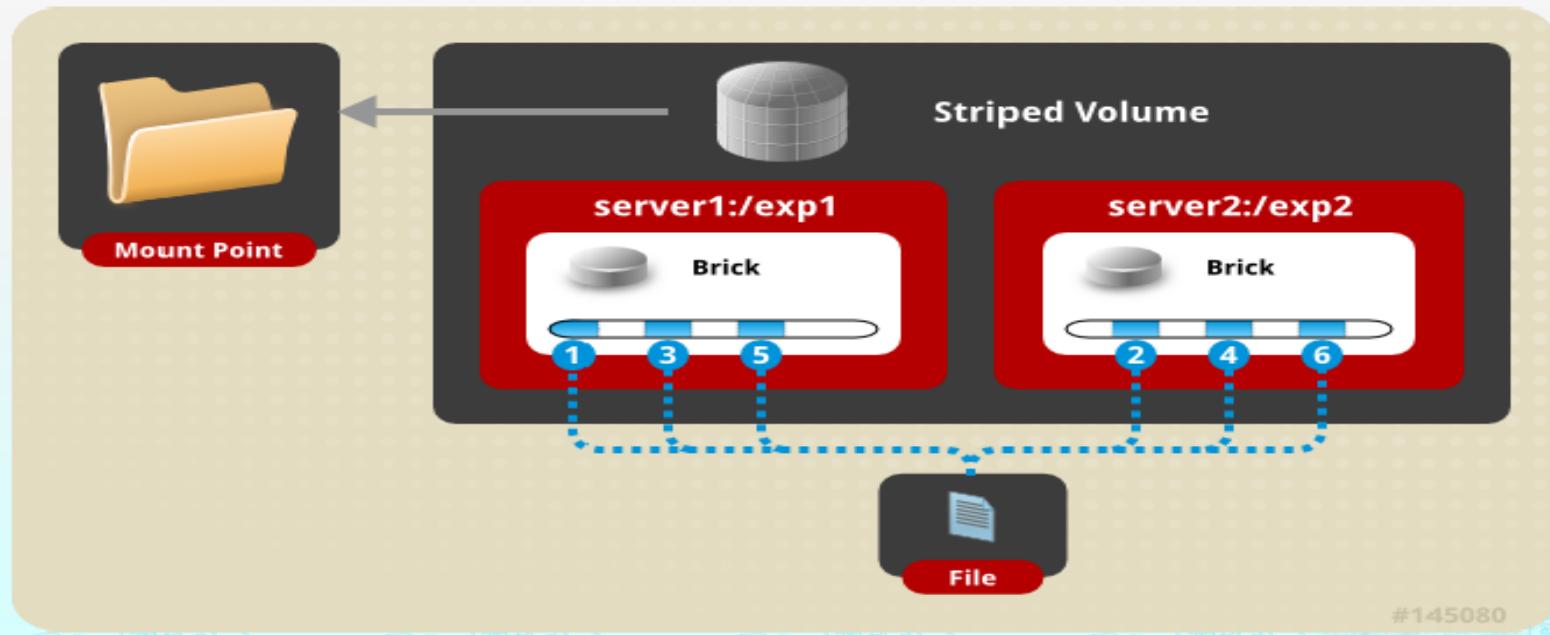
- 哈希卷和复制卷的复合方式
- 同时具有哈希卷和复制卷的特点



#145082

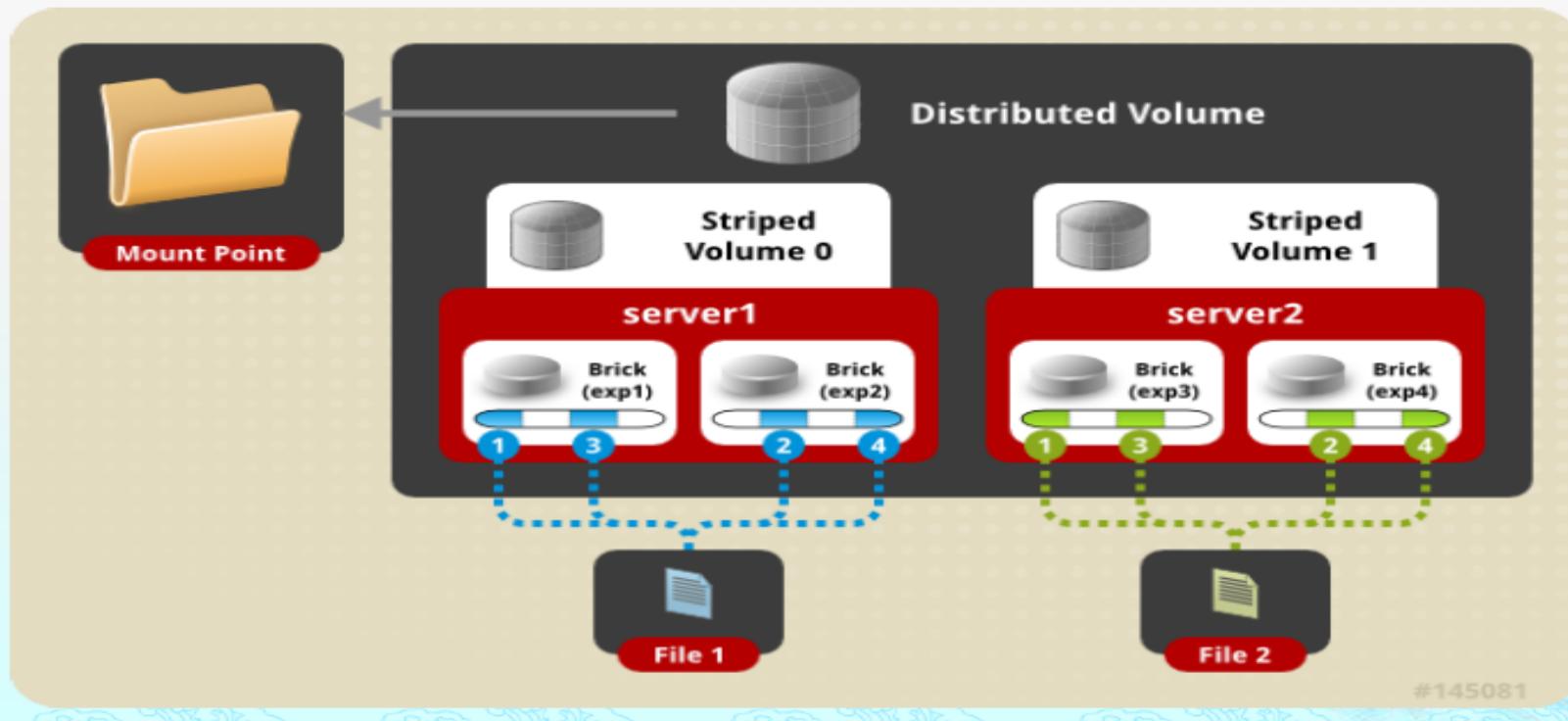
条带卷 (Striped Volumes)

- 单个文件分布到多个brick上，支持超大文件
- 类似RAID 0，以Round-Robin方式
- 通常用于HPC中的超大文件高并发访问



复合卷：哈希 + 条带

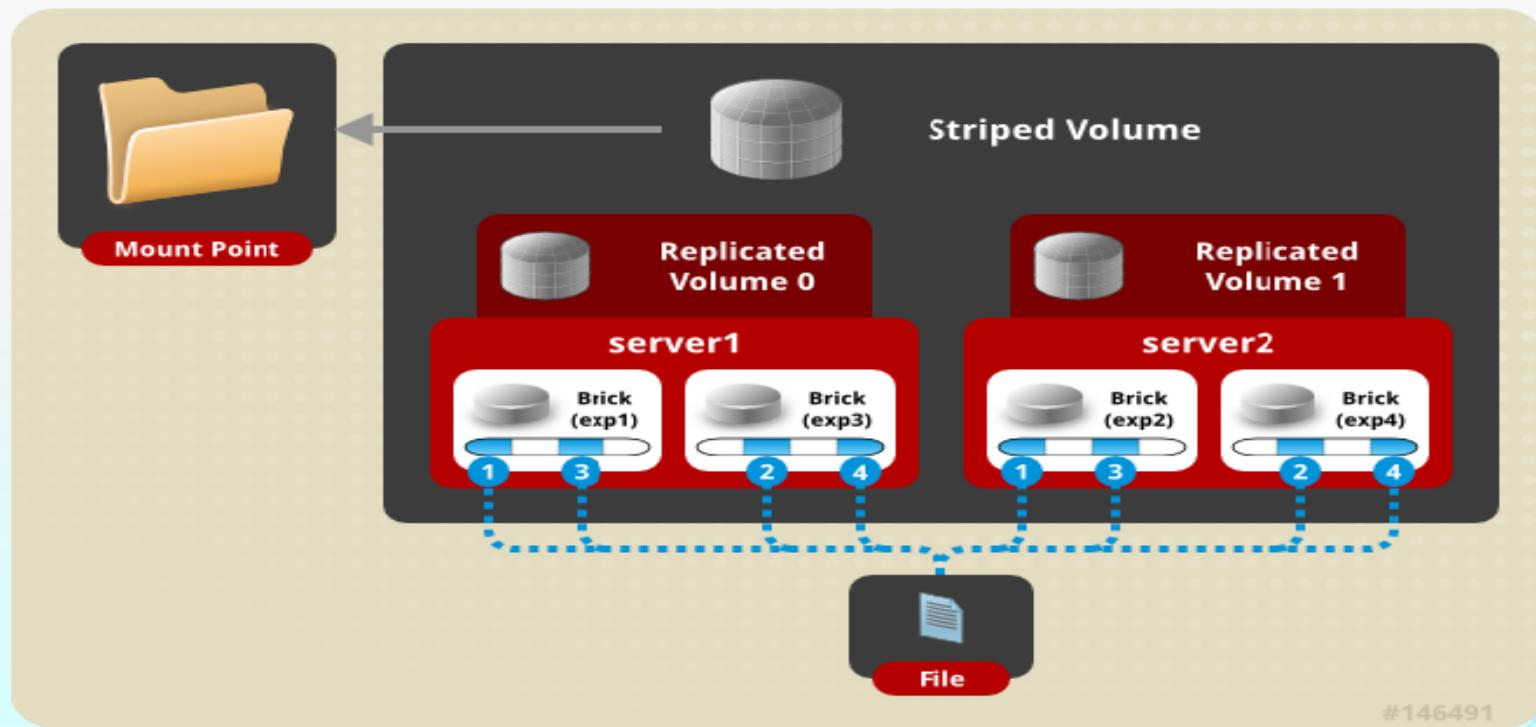
- 哈希卷和条带卷的复合方式
- 同时具有哈希卷和条带卷的特点



#145081

复合卷：条带 + 复制

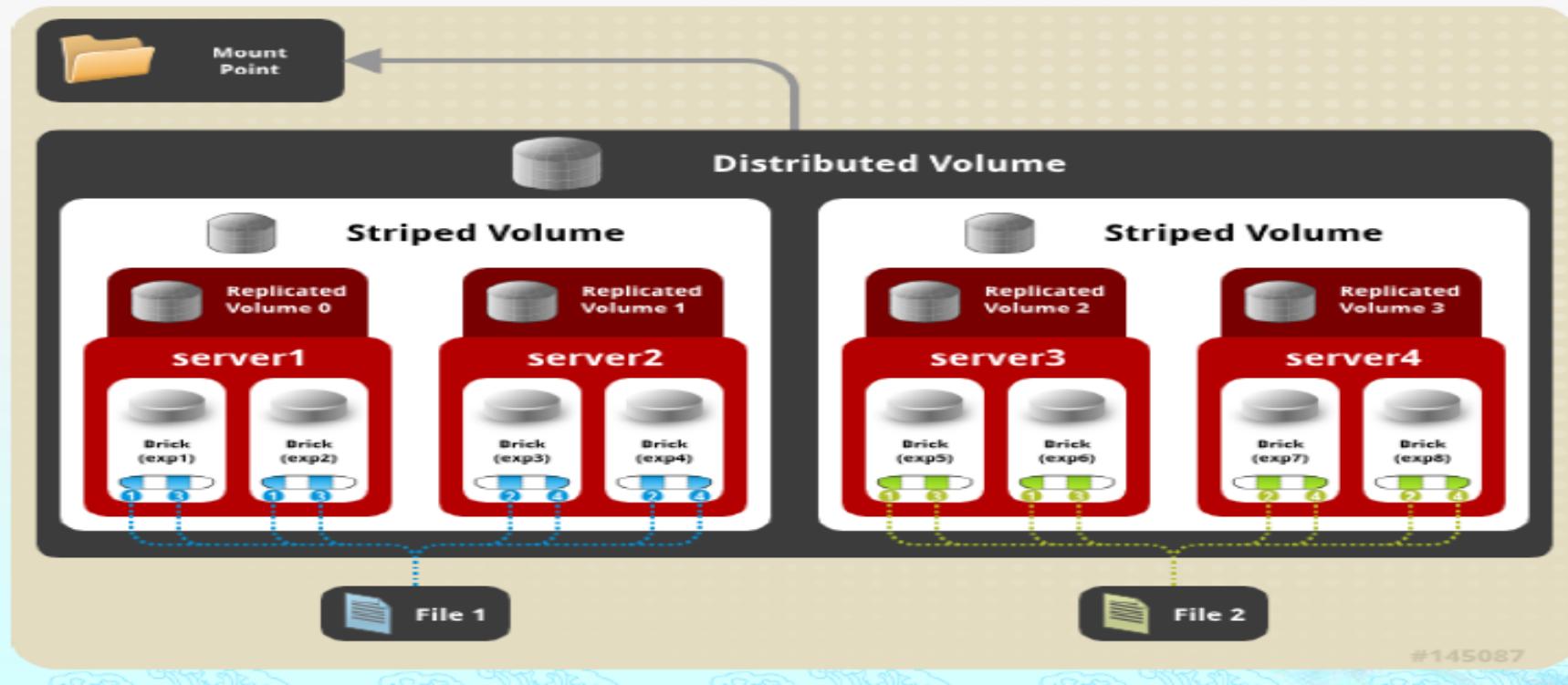
- 类似RAID 10
- 同时具有条带卷和复制卷的特点



#146491

复合卷：哈希 + 条带 + 复制

- 三种基本卷的复合卷
- 通常用于类Map Reduce应用



#145087

5、GlusterFS安装

安装的是glusterfs-3.6.0版本，安装环境是CentOS7，具体安装如下：

```
# tar -xzvf glusterfs-3.6.0.tar.gz  
# cd ./glusterfs-3.6.0  
# yum -y install openssl-devel  
# yum -y install libxml2-devel  
# ./configure  
# make  
# make install
```

下载网址

是：“<http://download.gluster.org/pub/gluster/glusterfs/3.6/3.6.0/>”

6、GlusterFS 常用translators（中继）

(1)“storage posix”

其作用就是指定一个本地目录给glusterfs内的一个卷使用，也就是在本地声明了一个目录卷给glusterfs。举例如下：

```
volume posix1
type storage posix
option directory /tmp/mydir
end-volume
```

(2) "features/locks"

锁中继的作用就是给服务端自己所开放的本地目录卷提供加锁功能。所以，锁中继只能在posix的后面，用来给卷加锁，不能在其他地方，举例如下：

volume locks

type features/locks

subvolumes posix1 posix2 (就是指定上锁的卷)

end-volume

(3) "performance/io-threads"

IO线程中继，是个转换器，作用就是增加**io**的并发线程，提高**io**功能，因为**glusterfs**服务是单线程，使用**IO**线程转换器可以较大地提高性能，这个转换器最好是被用于服务器端，而且是在服务器协议转换器后面被加载。**IO**线程操作会将读和写操作分成不同的线程，同一时刻存在的线程是恒定不变的而且是可以配置的。

volume iothreads

type performance/io-threads

option thread-count 32 (声明线程数量)

subvolumes locks

end-volume

(4) " protocol/server"

该服务器中继表示该节点在glusterfs中作为服务器模式.

```
volume server1
type protocol/server
option transport-type tcp
subvolumes brick1
option auth.addr.brick.allow *
end-volume
```

(5)“protocol/client”

客户端中继，用于客户端连接服务器时使用。

volume client1

type protocol/client

option transport-type tcp

option remote-host 192.168.103.49

option remote-port 6996

option remote-subvolume brick

end-volume

(6)“cluster/replicate”

复制中继，给glusterfs提供了类似RAID-1的功能，会复制文件或文件夹到各个subvolumes里面。

volume replicate1

type cluster/replicate

subvolumes brick1 brick2

end-volume

(7)“cluster/distribute”

分布式中继，给glusterfs提供了类似RAID-0的功能。可以把多个卷或子卷组合成一个大卷，实现多存储空间的聚合

volume distribute1

type cluster/distribute

subvolumes replicate1 replicate2

end-volume

(8)“performance/read-ahead”

预读中继，属于性能调整中继中的一种，用预读方式提高读取性能。预读转换器在每次读取操作前就预先抓取数据。这个有利于应用频繁持续性的访问文件，当应用完成当前数据块读取的时候，下一个数据块就已经准备好了。预读最好被使用在使用InfiniBand卡（或使用ib-verbs传输）的系统上。在快速以太网或者千兆以太网络环境中，就算不使用预读，Glusterfs也可以达到网卡最大连接的吞吐量，所以使用预读配置就是多余的。需要注意的是，预读操作只会发生在读的请求是完全连续的。如果应用访问数据很随机，那使用预读实际上将造成性能的损失，因为预读操作会拿一些应用并不会用到的数据块。

volume readahead

type performance/read-ahead

option page-size 256 （每次预读取数据块大小）

option page-count 4 (每次预读数据块数量)

#option force-atime-update （一般不用）

subvolumes XXXXXX

end-volume

(9)“performance/write-behind”

回写中继,属于性能调整中继的一种,作用是在写数据时,先写入缓存,在写入硬盘,以提高写入性能。

volume writebehind

type performance/write-behind

option cache-size 3MB (缓存大小)

option flush-behind on (适用于大量小文件)

subvolumes XXXXXXXX

end-volume

(10)“performance/io-cache”

缓存中继，属于性能调整中继的一种，作用是缓存已被读取过的数据，提高IO性能，适用于多个应用对同一数据多次访问，并且读的操作远远大于写的操作时是很有用的

volume iocache

type performance/io-cache

option cache-size 32MB (缓存的最大数据量)

option cache-timeout 1(验证超时时间)

#option priority (文件匹配列表及其优先级)

subvolumes XXXXXX

end-volume

除了以上10种中继还有其他的一些中继

cluster/nufa,cluster/stripe,cluster/ha,features/filter,features/trash等