# GlusterFS 与 Ceph 性能测试报告

-在相同网络、宿主机与存贮方式环境下文件读写性能比较测试

V3-版本改进：

- 确保 Gluster 的数据存贮虚拟机分布在不同的宿主机上；
- GlusterFS 存贮服务器增加了 2 块磁盘;
- Striped3+replica3 的 sr33 卷由各服务器的 vdb 构成；
- volume+replica3 的 sr33 卷由各服务器的 vdc 构成;
- Glusterfs 磁盘文件系统用 xfs;
- 给出了 GlusterFS 2 种存贮模式的读写数据报告;
- 增加了 GlusterFS 128K 时的读写数据报告
- Iozone 命令加入 I 参数据，免去客户端本身内存与硬盘间数据缓存对读写测试结果的影响，回避 V2 测试中 Ceph 出现的超过 1G 网卡最高数据吞吐量的情形，这样可以真实地获得后端文件系统实际的能力数据。当然正常使用时客户端访问后端文件系统时，客户端本身的缓存仍会起作用。因此这个 directio 测试只为查验而使用。
- 增加了利用 cp 命令从客户端直接向 GlusterFS,Ceph 复制文件的时间比较
- 增加了利用 cp 命令从客户端直接向 GlusterFS,Ceph 复制大量小文件的时间比较

成都信立讯科技有限公司

2014/3/10

目录

# 前言

GlusterFS 与 Ceph 是不一样的存贮系统，GlusterFS 更加针对分布式文件系统，虽然目前开发者也正在努力增加块与对象存贮服务。

由于两者在设计理念等各个方面有所不同，单纯的测试比较从功能应用的角度来说意义不大（如人们需要块及对象存贮时，目前 GlusterFS 只能部分提供或没有，块存贮也只能用于非生产环境，对象存贮还没有），但很多人使用时均会考虑将此两者文件服务作为其中一个重要应用，而同时，市场上流传着各类关于 GlusterFS 与 Ceph 对比的各类性能数据，实际情况倒底如可，我们不得而知，但我确实想知道真实的情况如何，或者大概的也行。

为此，我在同样的网络与硬件环境下，搭起了 2 套系统，使用了同样数量与配置的虚拟存贮服务器构成集群，同时也采用了同样的后台数据存贮方式，即 ceph 与 GlusterFS 均采用条带化与 3 份复本创建存贮池与卷，意即将需要存贮的文件完全切块并分散地存贮于不同的服务器中（3 台主服务器，6 台副本服务器）。

由于宿主机、存贮服务器、网络与存贮模式基本相同，因此此测试报告给出的数据有参考价值。另外，在测试中，并没有使用 Ceph 本身推荐的 brtf 文件系统，仍采用 xfs，但从 Inktank 的测试中表明，若使用 brtf, Ceph 集群的读写速度会至少增加 1 倍；相应地，GlusterFS 也是以 xfs 作为存贮服务器外挂磁盘的文件系统。

Ceph 社区-关于 Ceph 使用 brtf、xfs、ext3 文件系统的读写性能比较：
http://ceph.com/community/ceph-bobtail-performance-io-scheduler-comparison/

关于 brtf 文件系统能否用于生产系统的文章（意思是可用），但目前不知谁在用：
http://www.oschina.net/news/46450/btrfs-stable

在于 brtf 文件系统的简要介绍：
http://www.oschina.net/p/btrfs

# 测试环境主机与网络背景说明

我们采用虚拟机的安装方式，在同一套环境中分别部署了 GlusterFS 与 Ceph 的虚拟机，结构

图如下，每个宿主机 Intel 主板（双路 8 核），64G 内存，用 2 块 2T SATA2 硬盘（虚拟机都创建在第一块硬盘上），4 块 1G 网卡(创建 4 个网桥，每台虚拟机 4 张网卡分别挂到不同的桥上), GlusterFS 与 Ceph 的网络环境完全相同；宿主机操作系统为 Centos 6.4 64 位。

所有的 Ceph,GlusterFS 的存贮服务与客户端均是 2G 内存。

所有测试结节点网卡均用 ethtool 检验过，均是 1000M 速率连接，例：

```
[root@storage2 ~]# ethtool eth3
Settings for eth3:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: Symmetric
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: Symmetric
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: Unknown
        Supports Wake-on: pumbg
        Wake-on: d
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

# 测试环境说明

# Gluster 与 Ceph 在数据存贮模式的差异

Ceph 的所有数据在存贮系统中均以切块保存（object），每个文件对应系统的某个存贮池，在创建存贮池时，可以指定该池文件需要保存的数量，比如我们创建一个 Test 池，此池所有的文件保存三份：

ceph osd pool create test 128
ceph osd pool set test size 3

Ceph 存贮池中的文件映射到不同的 object 中，然后再放入不同的 PG 之中，而不同的 PG 又分散存贮于存贮系统的所有服务器之中，增加或减少存贮服务器数量，系统会自动完成平衡，不需要人为干预。当选择为 3 份副本后，允许的最大情况是，系统同时有 2 台服务器损坏时，不会丢失数据。

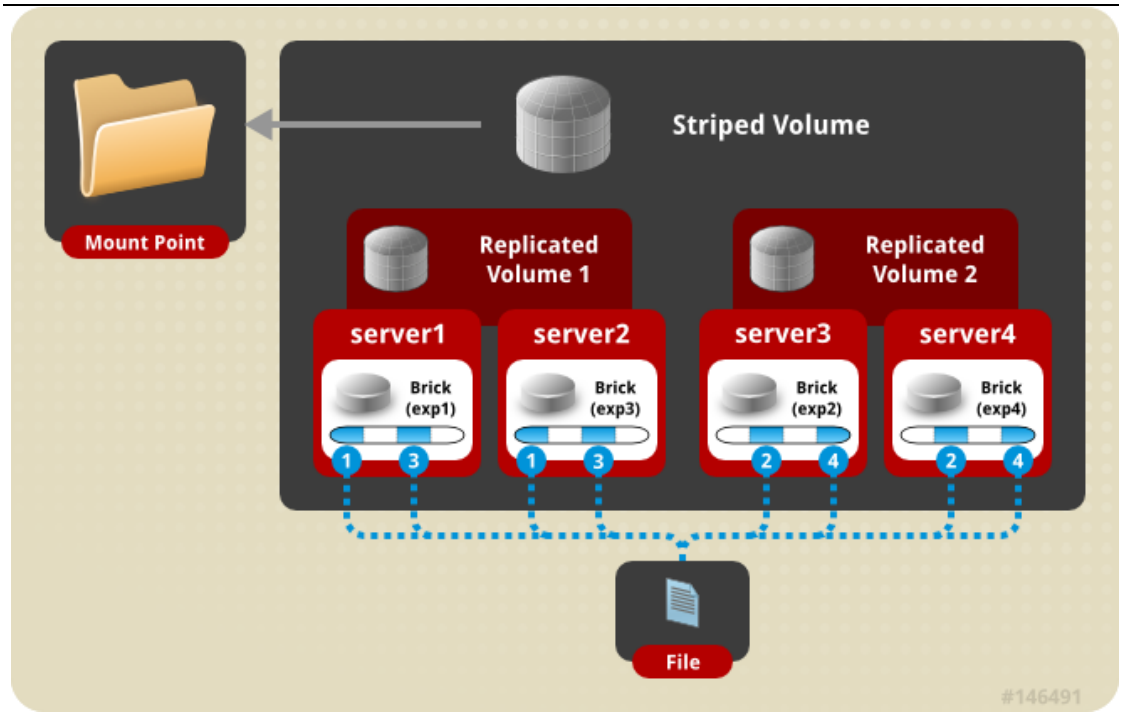
GlusterFS 创建条带化（striped）存贮卷与数据存放三份的实现模式与 Ceph 略有不同。Redhat 的官方文档为：
https://access.redhat.com/site/documentation/en-US/Red_Hat_Storage/2.0/html/Administration_Guide/sect-User_Guide-Setting_Volumes-Striped_Replicated.html

比如我们创建一个分散在 2 台服务器中、数据 2 份副本，允许损坏一台服务器而数据不丢失的命令是：

```
    # gluster volume create sr22 stripe 2 replica 2 transport tcp
server1:/exp2 server2:/exp4 server3:/exp2 server4:/exp4
```

意即数据存放到哪些服务器中由人工指定，而不是系统自动完成，这在大规模存贮系统中有可能有点麻烦。

存贮的模式是，文件的第 1，3 块存贮于 server1 中，2，4 块存贮于 server2 中，因为数据需要保留 2 份，因此还要有 2 台服务器用于与 server1, server2 完成镜像，如 server3, server4.




如果我们需要将数据条带分散到 3 台服务器，同时文件存贮 3 份，那么则需要 9 台服务器才能完成，此时才能达到同时损坏 2 台服务器数据不丢失的能力。

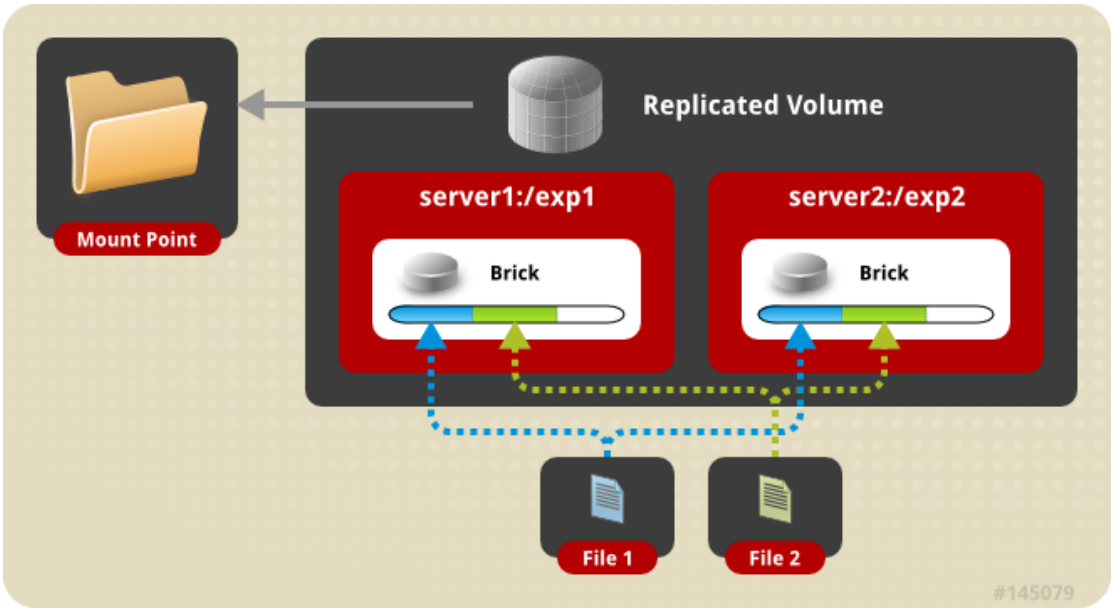
当然，我们还有一种方案是数据条带化到 1 台服务器中，数据存贮三份，这样只需要 3 台服务器就可以满足要求，但问题是 1 台服务器的网络吞吐与性能无法满足大量访问要求，因此此模式不合适。

GlusterFS 集群中如果新增服务器，则需要手工进行平衡（Rebalance），如果不做均衡，那么新的数据仍会存贮于原有的服务器中。

## 测试中采用的 Ceph 与 GlusterFS 的存贮模式

GlusterFS 还有一种以文件为基础不切块存放的副本（Replicated）模式，如下图所示：

Redhat 的官方文档：
https://access.redhat.com/site/documentation/en-US/Red_Hat_Storage/2.0/html/Administration_Guide/sect-User_Guide-Setting_Volumes-Replicated.html）




GlusterFS 采用文件存贮模式与 Ceph 采用文件切块模式有着本质上的不同，如果这样测试不对等，得出的数据也没有十足的意义，因此本次测试中我们采用了文件切块模式，即 GlusterFS 采用 Striped 3 Replica 3 的存贮模式创建卷，而 Ceph 也创建 Replica 为 3 的存贮池；GlusterFS 的文件会分散存贮于 9 台服务器中，Ceph 的存贮池也由 128 个 PG 构成，也会分散于 9 个台服务器中。因此，从整体上看，这样的对比测试得出的数据还有些参考价值。

## Ceph 测试环境说明

以下是 Ceph 的测试环境，说明如下：

- Ceph 采用 0.72 版本，安装于 Ubuntu 12.04 LTS 版本中；系统为初始安装，没有调优。
- 每个 OSD 虚拟存贮服务器 2 个 VCPU,2G 内存，挂载 4 块 100G 虚拟磁盘；第一块用于操作系统；第二块用于日志，另 2 块用于数据存贮,即每个 OSD 虚拟存贮服务器运行 2 个 ceph-osd 进程：

```
root@cephmona:~# ceph osd tree
# id    weight  type name       up/down reweight
-1      1.8     root default
-14     1.8             datacenter dc1
-13     1.8                     room room1
-12     1.8                             row row1
```

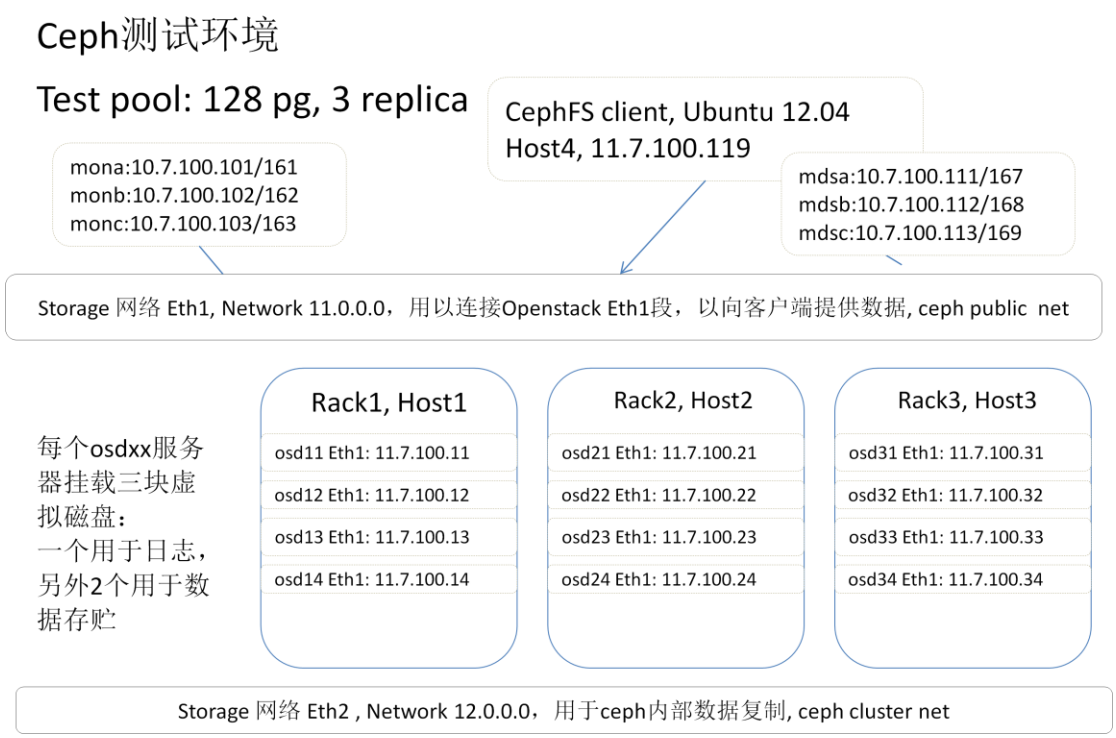
```
-11     0                       rack rack1
-15     0                       rack rack2
-16     0                       rack rack3
-17     0.5999                  rack rack4
-2      0.2                             host cephosd11
0       0.09999                                 osd.0   up      1
1       0.09999                                 osd.1   up      1
-3      0.2                             host cephosd12
2       0.09999                                 osd.2   up      1
3       0.09999                                 osd.3   up      1
-4      0.2                             host cephosd13
4       0.09999                                 osd.4   up      1
5       0.09999                                 osd.5   up      1
-18     0.5999                  rack rack5
-5      0.2                             host cephosd21
6       0.09999                                 osd.6   up      1
7       0.09999                                 osd.7   up      1
-6      0.2                             host cephosd22
8       0.09999                                 osd.8   up      1
9       0.09999                                 osd.9   up      1
-7      0.2                             host cephosd23
10      0.09999                                 osd.10  up      1
11      0.09999                                 osd.11  up      1
-19     0.5999                  rack rack6
-8      0.2                             host cephosd31
12      0.09999                                 osd.12  up      1
13      0.09999                                 osd.13  up      1
-9      0.2                             host cephosd32
14      0.09999                                 osd.14  up      1
15      0.09999                                 osd.15  up      1
-10     0.2                             host cephosd33
16      0.09999                                 osd.16  up      1
17      0.09999                                 osd.17  up      1
```

- 使用 Test pool，此池为 128 个 PGs,数据存三份；

```
root@cephmona:~# ceph osd pool get test size
size: 3
root@cephmona:~# ceph osd pool get test pg_num
pg_num: 128
root@cephmona:~#
```

- Ceph osd 采用 xfs 文件系统（若使用 brtf 文件系统读写性能将翻 2 倍）；

- 客户端运行在 Host4 上，使用 11.x.x.x 网段访问 Ceph 存贮系统的 public 网段进行数据读写；
- Ceph 系统中的块采用默认安装，为 64K




## GlusterFS 测试环境说明

GlusterFS 环境如下：

- GlusterFS 安装于 Centos 6.4 64 位版本，GlusterFS 版本为 3.4.2。系统为初始安装，没有调优。
- GlusterFS 由 9 台虚拟服务器构成，每台服务器 2 个 VCPU,2G 内存，挂载三块 40G 磁盘，一块用于操作系统，另二块用于 brick 数据存贮，集群情况如下：

在 brick11 存贮服务器上可以看到其它 8 个节点：
[root@brick11 ~]# gluster peer status

```
Number of Peers: 8
```

```
Hostname: brick12
Uuid: 3668b021-7109-4670-a781-53b0b978a280
State: Peer in Cluster (Connected)

Hostname: brick31
Uuid: 6a2ff7b9-bca0-4d81-ad45-48c71b1e02eb
State: Peer in Cluster (Connected)

Hostname: brick13
Uuid: b91dc2d3-ced0-4c9e-a7ed-1803cb2578e0
State: Peer in Cluster (Connected)

Hostname: brick23
Uuid: 81c187a8-a6e7-4c7e-a385-cd62425aaec1
State: Peer in Cluster (Connected)

Hostname: brick21
Uuid: 86f4a8fb-f13f-4262-8ba5-31614f80df50
State: Peer in Cluster (Connected)

Hostname: brick32
Uuid: bdca294f-8c04-4e2f-a8fc-05458035ce82
State: Peer in Cluster (Connected)

Hostname: brick33
Uuid: 0cb4779a-4b57-4c95-b606-5fb9f6c5189a
State: Peer in Cluster (Connected)

Hostname: brick22
Uuid: d045173f-cb31-47d1-a5e9-0f768f0767ab
State: Peer in Cluster (Connected)
```

- GlusterFS 每个 brick 挂载的磁盘采用 xfs 文件系统；
- GlusterFS 通过 10.x.x.x 的 IP 地址创建集群；
- **在 GlusterFS 上创建 striped 3, replica 3 的 sr33 卷以及 volume replica 3 的 vr33 卷(V3 版本特别处理)，此 2 数据卷放在不同的磁盘上：**

```
gluster volume create sr33 stripe 3 replica 3 transport tcp brick11:/mnt/sr/sr11 brick21:/mnt/sr/sr21 brick31:/mnt/sr/sr31 brick12:/mnt/sr/sr12 brick22:/mnt/sr/sr22 brick32:/mnt/sr/sr32 brick13:/mnt/sr/sr13 brick23:/mnt/sr/sr23 brick33:/mnt/sr/sr33

再创建一个以文件为基础的保存于三台机器中的卷：
```

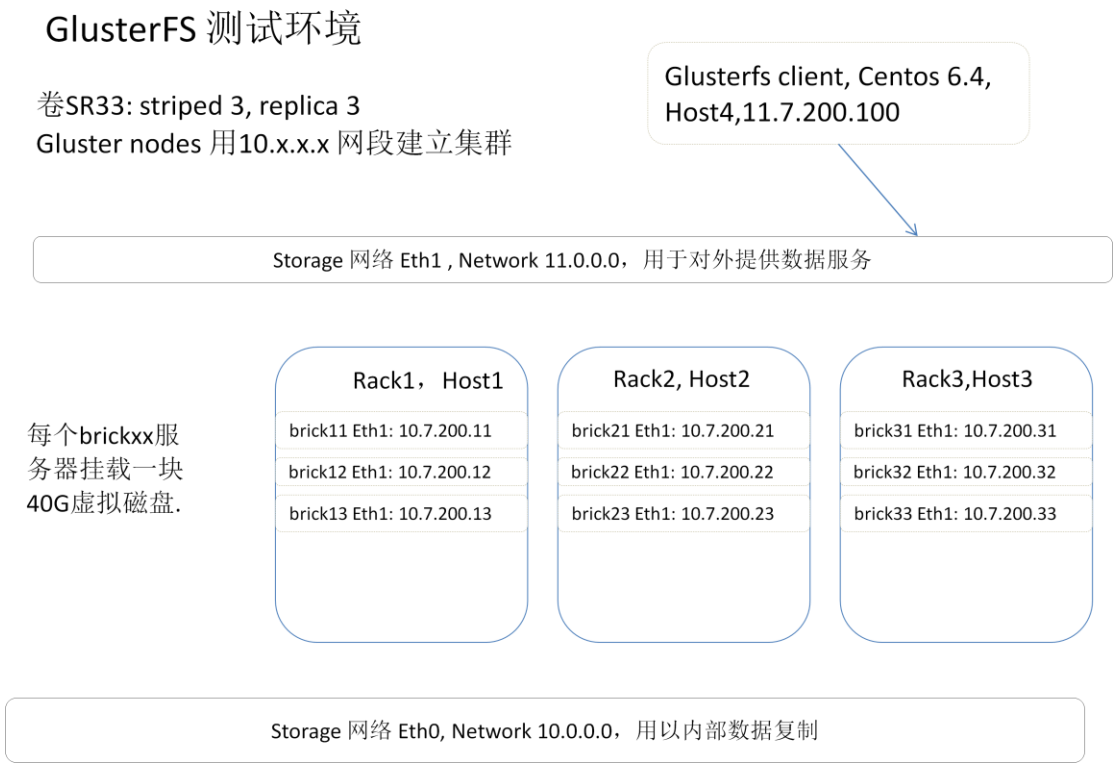
```
gluster volume create vr33 replica 3 transport tcp brick11:/mnt/vr/vr11 brick21:/mnt/vr/vr21 brick31:/mnt/vr/vr31 brick12:/mnt/vr/vr12 brick22:/mnt/vr/vr22 brick32:/mnt/vr/vr32 brick13:/mnt/vr/vr13 brick23:/mnt/vr/vr23 brick33:/mnt/vr/vr33


[root@brick11 ~]# gluster volume start sr33
volume start: sr33: success
[root@brick11 ~]# gluster volume start vr33
volume start: vr33: success
[root@brick11 ~]#
```

- **卷的创建已充分考虑了虚拟机的不同，即将卷创建到不同的物机的虚拟机中（V3 版本特别处理）；**
- 客户端仍使用 10.x.x.x 网段访问 GlusterFS 集群；客户端运行在 Host4 上：

```
[root@gluserclient /]# mount -t glusterfs brick11:/sr33 /mnt/sr33
[root@gluserclient /]# mount -t glusterfs brick11:/vr33 /mnt/vr33
```

- GlusterFS 中的块采用默认安装，为 128K.




GlusterFS 集群使用的存贮模式

https://access.redhat.com/site/documentation/en-US/Red_Hat_Storage/2.0/html/Administration_Guide/sect-User_Guide-Setting_Volumes-Striped_Replicated.html

# 测试工具与方法说明

## 测试工具 iozone 说明

测试工具采用 iozone 3.4；

关于 iozone 各项参数的说明，摘抄于网上：
http://blog.sina.com.cn/s/blog_3cba7ec10100ea62.html

另外，这里文章揭示出使用 iozone 回避磁盘缓存的一些方法：
http://blog.chinaunix.net/uid-20196318-id-28764.html

测试时，ceph 与 Gluster 均停止所有其他外部访问，并且依次进行测试。先测试 GlusterFS 集群，再测试 Ceph 集群。

## 测试方法及影响说明

## 考虑实际应用中 linux 本身内存与硬盘间的缓存的情况

分别用 4,16,64K 三种块大小进行 128M-1G 文件大小（以 128M 大小递增）的写、读及随机读写，生成表格数据 ，命令如下：

Iozone 命令如下：
[root@gluserclient ~]# **iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -f /mnt/gluster/iozonetest |tee ./test-write-read.log &**

注：由于考虑到 Ceph 系统的默认安装文件读取块大小是 64K,因此最大只测到 64K，而 GlusterFS 的默认安装为 128K，若测 128K,Ceph 存贮服务器需要读 2 次，担心数据经不起检验，因此，没有测 128K 的情况。

## 跳过客户端 linux 内存与硬盘间缓存的情况（V3 版本改进）

由于 linux 页高速缓存机制即延迟写机制，写文件时数据先写入高速缓存，在内存不足或是一定时间之后会被刷新到磁盘。读文件时，先检查文件数据是否在高速缓存中，如果在则直接从缓存中读取；否则从磁盘读取至高速缓存。

这样，我们获得的数据并不能真实地反应 GlusterFS 与 Ceph 系统的真实性能，受客户端本机影响较大（客户端有 2G 内存，而我们测试的最大文件为 1G）.

为了更好地查看文件系统的性能，我们需要消除操作系统本身缓存的影响，因此，我们需要进一步在 iozone 中加入 –I 选项，以便查看真实的 GlusterFS 及 Ceph 提供的文件服务的读写

速率。

Iozone 的 I 选项意义：

-I ： 对所有文件操作使用 DIRECT I/O。通知文件系统所有操作跳过缓存直接在磁盘上操作

这也有可能回避前期测试中 ceph 可以获得超过网卡线速的不可思议的测试数据。

此时所使用的 iozone 命令为：

iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K **-I** -f /mnt/sr33/iozonetest |tee ./test-write-read.log &

# GlusterFS 的二种测试模式及使用或不使用客户端缓存机制及数据

以下的测试，我们对 GlusterFS 进行了 2 种测试，一种是与 Ceph 对应用 stiped+replica 模式，另一种是 volume+replica 的模式。

我们标记存在缓存时为 Cached,起用 iozone direct io 时， 标记为 Non-Cached

## GlusterFS 集群的测试 log-striped3+replica3-Cached

[root@gluserclient ~]# **iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -f /mnt/sr33/iozonetest |tee ./test-write-read.log &**

先进行 glusterclient 对 glusterFS 集群的读写测试， log 如下：

[root@gluserclient ~]# cat test-write-read.log

```
        Iozone: Performance Test of File I/O
                Version $Revision: 3.408 $
                Compiled for 64 bit mode.
                Build: linux

        Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                        Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                        Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                        Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                        Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                        Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
```

Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer.
Ben England.

Run began: Tue Mar 11 18:17:54 2014

Excel chart generation enabled
Using minimum file size of 131072 kilobytes.
Using maximum file size of 1048576 kilobytes.
Record Size 4 KB
Record Size 16 KB
Record Size 64 KB
Command line used: iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -f /mnt/sr33/iozonetest
Output is in Kbytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 Kbytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | random | random | bkwd | record | stride | |
| | KB | reclen | write | rewrite | read | reread | read | write | read | rewrite | read | fwrite frewrite fread freread |
| | 131072 | 4 | 16996 | 15086 | 114616 | 114704 | 7715 | 8095 | | | | |
| | 131072 | 16 | 36216 | 36251 | 114708 | 114705 | 21225 | 25301 | | | | |
| | 131072 | 64 | 38471 | 38477 | 114635 | 114684 | 44325 | 38433 | | | | |
| | 262144 | 4 | 20557 | 20710 | 114615 | 114728 | 6499 | 8727 | | | | |
| | 262144 | 16 | 37576 | 35912 | 114708 | 114718 | 19611 | 25132 | | | | |
| | 262144 | 64 | 38328 | 38330 | 114704 | 114705 | 43109 | 38321 | | | | |
| | 524288 | 4 | 19979 | 17783 | 114173 | 114725 | 6135 | 8531 | | | | |
| | 524288 | 16 | 35918 | 36322 | 114391 | 114750 | 18559 | 29505 | | | | |
| | 524288 | 64 | 36743 | 38260 | 114735 | 114722 | 41879 | 38253 | | | | |
| | 1048576 | 4 | 20240 | 18153 | 114528 | 114624 | 5891 | 8576 | | | | |
| | 1048576 | 16 | 36457 | 37048 | 114548 | 114663 | 17795 | 32901 | | | | |
| | 1048576 | 64 | 37802 | 38226 | 114747 | 114754 | 41421 | 38224 | | | | |

iozone test complete.
Excel output is below:

"Writer report"
"4" "16" "64"
"131072" 16996 36216 38471
"262144" 20557 37576 38328

```
"524288"     19979    35918    36743
"1048576"     20240    36457    37802

"Re-writer report"
            "4"    "16"    "64"
"131072"     15086    36251    38477
"262144"     20710    35912    38330
"524288"     17783    36322    38260
"1048576"     18153    37048    38226

"Reader report"
            "4"    "16"    "64"
"131072"     114616    114708    114635
"262144"     114615    114708    114704
"524288"     114173    114391    114735
"1048576"     114528    114548    114747

"Re-Reader report"
            "4"    "16"    "64"
"131072"     114704    114705    114684
"262144"     114728    114718    114705
"524288"     114725    114750    114722
"1048576"     114624    114663    114754

"Random read report"
            "4"    "16"    "64"
"131072"     7715    21225    44325
"262144"     6499    19611    43109
"524288"     6135    18559    41879
"1048576"     5891    17795    41421

"Random write report"
            "4"    "16"    "64"
"131072"     8095    25301    38433
"262144"     8727    25132    38321
"524288"     8531    29505    38253
"1048576"     8576    32901    38224
```

# GlusterFS 集群的测试 log-striped3+replica3-Non-Cached

[root@gluserclient ~]# **iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -I -f /mnt/sr33/iozonetest |tee ./test-write-read.log &**

```
[root@gluserclient ~]# cat test-write-read.log
        Iozone: Performance Test of File I/O
                Version $Revision: 3.408 $
                Compiled for 64 bit mode.
                Build: linux

        Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                        Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                        Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                        Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                        Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                        Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                        Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer.
                        Ben England.

        Run began: Wed Mar 12 11:10:09 2014

        Excel chart generation enabled
        Using minimum file size of 131072 kilobytes.
        Using maximum file size of 1048576 kilobytes.
        Record Size 4 KB
        Record Size 16 KB
        Record Size 64 KB
        O_DIRECT feature enabled
        Command line used: iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r
16K -r 64K -I -f /mnt/sr33/iozonetest
        Output is in Kbytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 Kbytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
                                                                random          random
bkwd    record  stride
                KB  reclen   write  rewrite    read    reread    read   write    read
rewrite     read   fwrite frewrite   fread  freread
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 131072 | 4 | 11943 | 21903 | 3626 | 3629 | 2359 | 7714 |
| 131072 | 16 | 29236 | 35147 | 13449 | 13622 | 8972 | 32530 |
| 131072 | 64 | 33728 | 35434 | 38100 | 37949 | 29738 | 33872 |
| 262144 | 4 | 13807 | 19169 | 3585 | 3620 | 2106 | 8240 |
| 262144 | 16 | 34259 | 23590 | 13491 | 13673 | 8088 | 23896 |
| 262144 | 64 | 38237 | 26697 | 38124 | 37996 | 28778 | 30723 |
| 524288 | 4 | 9998 | 17658 | 3558 | 3580 | 1997 | 7625 |
| 524288 | 16 | 18342 | 22729 | 13397 | 13476 | 7738 | 25647 |
| 524288 | 64 | 33110 | 33544 | 37926 | 37743 | 27817 | 30123 |
| 1048576 | 4 | 11813 | 12283 | 3605 | 3596 | 1952 | 2411 |
| 1048576 | 16 | 16156 | 19633 | 13407 | 13431 | 7575 | 24366 |
| 1048576 | 64 | 26658 | 27618 | 37915 | 38069 | 27430 | 19191 |

iozone test complete.
Excel output is below:

"Writer report"
"4" "16" "64"
"131072" 11943 29236 33728
"262144" 13807 34259 38237
"524288" 9998 18342 33110
"1048576" 11813 16156 26658

"Re-writer report"
"4" "16" "64"
"131072" 21903 35147 35434
"262144" 19169 23590 26697
"524288" 17658 22729 33544
"1048576" 12283 19633 27618

"Reader report"
"4" "16" "64"
"131072" 3626 13449 38100
"262144" 3585 13491 38124
"524288" 3558 13397 37926
"1048576" 3605 13407 37915

"Re-Reader report"
"4" "16" "64"
"131072" 3629 13622 37949
"262144" 3620 13673 37996
"524288" 3580 13476 37743

```
"1048576"     3596    13431    38069

"Random read report"
            "4"    "16"    "64"
"131072"     2359    8972    29738
"262144"     2106    8088    28778
"524288"     1997    7738    27817
"1048576"     1952    7575    27430

"Random write report"
            "4"    "16"    "64"
"131072"     7714    32530    33872
"262144"     8240    23896    30723
"524288"     7625    25647    30123
"1048576"     2411    24366    19191
```

## GlusterFS 集群的测试 log-volume+replica3-Cached

[root@gluserclient ~]# **iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -f /mnt/vr33/iozonetest |tee ./test-write-read.log &**

[root@gluserclient ~]# cat test-write-read.log

```
[root@gluserclient ~]# cat test-write-read.log
            Iozone: Performance Test of File I/O
                        Version $Revision: 3.408 $
                        Compiled for 64 bit mode.
                        Build: linux

            Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                                Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                                Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                                Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                                Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                                Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
```

Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer.
Ben England.

Run began: Tue Mar 11 18:51:14 2014

Excel chart generation enabled
Using minimum file size of 131072 kilobytes.
Using maximum file size of 1048576 kilobytes.
Record Size 4 KB
Record Size 16 KB
Record Size 64 KB
Command line used: iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -f /mnt/vr33/iozonetest
Output is in Kbytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 Kbytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | random | random | bkwd | record | stride | |
| KB | reclen | write | rewrite | read | reread | read | write | read | rewrite | read | fwrite | frewrite | fread | freread |
| 131072 | 4 | 19587 | 19242 | 114136 | 114688 | 8044 | 6852 | | | | | | | |
| 131072 | 16 | 38285 | 37644 | 114311 | 114639 | 21755 | 25480 | | | | | | | |
| 131072 | 64 | 38539 | 38543 | 114390 | 114737 | 45785 | 38444 | | | | | | | |
| 262144 | 4 | 19503 | 17628 | 114524 | 114715 | 6768 | 8957 | | | | | | | |
| 262144 | 16 | 37870 | 37885 | 114335 | 114392 | 19636 | 27465 | | | | | | | |
| 262144 | 64 | 38397 | 38405 | 114607 | 114720 | 43916 | 38322 | | | | | | | |
| 524288 | 4 | 20596 | 21649 | 114645 | 114562 | 6364 | 8756 | | | | | | | |
| 524288 | 16 | 38107 | 37970 | 114225 | 114733 | 18481 | 31535 | | | | | | | |
| 524288 | 64 | 38334 | 38331 | 114690 | 114697 | 42984 | 38260 | | | | | | | |
| 1048576 | 4 | 19803 | 20612 | 114537 | 114742 | 6105 | 7529 | | | | | | | |
| 1048576 | 16 | 37983 | 38052 | 114569 | 114715 | 17819 | 27388 | | | | | | | |
| 1048576 | 64 | 38297 | 38297 | 114742 | 114703 | 42440 | 38215 | | | | | | | |

iozone test complete.
Excel output is below:

"Writer report"
"4" "16" "64"
"131072" 19587 38285 38539
"262144" 19503 37870 38397

```
"524288"     20596    38107    38334
"1048576"     19803    37983    38297

"Re-writer report"
           "4"    "16"    "64"
"131072"     19242    37644    38543
"262144"     17628    37885    38405
"524288"     21649    37970    38331
"1048576"     20612    38052    38297

"Reader report"
           "4"    "16"    "64"
"131072"     114136    114311    114390
"262144"     114524    114335    114607
"524288"     114645    114225    114690
"1048576"     114537    114569    114742

"Re-Reader report"
           "4"    "16"    "64"
"131072"     114688    114639    114737
"262144"     114715    114392    114720
"524288"     114562    114733    114697
"1048576"     114742    114715    114703

"Random read report"
           "4"    "16"    "64"
"131072"     8044    21755    45785
"262144"     6768    19636    43916
"524288"     6364    18481    42984
"1048576"     6105    17819    42440

"Random write report"
           "4"    "16"    "64"
"131072"     6852    25480    38444
"262144"     8957    27465    38322
"524288"     8756    31535    38260
"1048576"     7529    27388    38215
```

# GlusterFS 集群的测试数据整理

GlusterFS 测试的数据,读写的数据单位为 Kbyte/sec-Striped3+replica3-Cached

| 文件大小<br>KB | 读写块<br>KB | 顺序写<br>Write | 顺序重写<br>rewrite | 顺序读<br>read | 顺序重读<br>reread | 随机读<br>random read | 随机写<br>random write |
|---|---|---|---|---|---|---|---|
| 131072 | 4 | 16996 | 15086 | 114616 | 114704 | 7715 | 8095 |
| 131072 | 16 | 36216 | 36251 | 114708 | 114705 | 21225 | 25301 |
| 131072 | 64 | 38471 | 38477 | 114635 | 114684 | 44325 | 38433 |
| 262144 | 4 | 20557 | 20710 | 114615 | 114728 | 6499 | 8727 |
| 262144 | 16 | 37576 | 35912 | 114708 | 114718 | 19611 | 25132 |
| 262144 | 64 | 38328 | 38330 | 114704 | 114705 | 43109 | 38321 |
| 524288 | 4 | 19979 | 17783 | 114173 | 114725 | 6135 | 8531 |
| 524288 | 16 | 35918 | 36322 | 114391 | 114750 | 18559 | 29505 |
| 524288 | 64 | 36743 | 38260 | 114735 | 114722 | 41879 | 38253 |
| 1048576 | 4 | 20240 | 18153 | 114528 | 114624 | 5891 | 8576 |
| 1048576 | 16 | 36457 | 37048 | 114548 | 114663 | 17795 | 32901 |
| 1048576 | 64 | 37802 | 38226 | 114747 | 114754 | 41421 | 38224 |

GlusterFS 测试的数据,读写的数据单位为 Kbyte/sec-Striped3+replica3-Non-Cached

| 文件大小<br>File | 读写块<br>block | 顺序写<br>s-srite | 顺序重写<br>s-rewrite | 顺序读<br>s-read | 顺序重读<br>s-reread | 随机读<br>r-read | 随机写<br>r-write |
|---|---|---|---|---|---|---|---|
| 131072 | 4 | 11943 | 21903 | 3626 | 3629 | 2359 | 7714 |
| 131072 | 16 | 29236 | 35147 | 13449 | 13622 | 8972 | 32530 |
| 131072 | 64 | 33728 | 35434 | 38100 | 37949 | 29738 | 33872 |
| 262144 | 4 | 13807 | 19169 | 3585 | 3620 | 2106 | 8240 |
| 262144 | 16 | 34259 | 23590 | 13491 | 13673 | 8088 | 23896 |
| 262144 | 64 | 38237 | 26697 | 38124 | 37996 | 28778 | 30723 |
| 524288 | 4 | 9998 | 17658 | 3558 | 3580 | 1997 | 7625 |
| 524288 | 16 | 18342 | 22729 | 13397 | 13476 | 7738 | 25647 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 524288 | 64 | 33110 | 33544 | 37926 | 37743 | 27817 | 30123 |
| 1048576 | 4 | 11813 | 12283 | 3605 | 3596 | 1952 | 2411 |
| 1048576 | 16 | 16156 | 19633 | 13407 | 13431 | 7575 | 24366 |
| 1048576 | 64 | 26658 | 27618 | 37915 | 38069 | 27430 | 19191 |

GlusterFS 测试的数据,读写的数据单位为 Kbyte/sec-Volume+replica3-Cached

| 文件大小<br>KB | 读写块<br>KB | 顺序写<br>Write | 顺序重写<br>rewrite | 顺序读<br>read | 顺序重读<br>reread | 随机读<br>random read | 随机写<br>random write |
|---|---|---|---|---|---|---|---|
| 131072 | 4 | 19587 | 19242 | 114136 | 114688 | 8044 | 6852 |
| 131072 | 16 | 38285 | 37644 | 114311 | 114639 | 21755 | 25480 |
| 131072 | 64 | 38539 | 38543 | 114390 | 114737 | 45785 | 38444 |
| 262144 | 4 | 19503 | 17628 | 114524 | 114715 | 6768 | 8957 |
| 262144 | 16 | 37870 | 37885 | 114335 | 114392 | 19636 | 27465 |
| 262144 | 64 | 38397 | 38405 | 114607 | 114720 | 43916 | 38322 |
| 524288 | 4 | 20596 | 21649 | 114645 | 114562 | 6364 | 8756 |
| 524288 | 16 | 38107 | 37970 | 114225 | 114733 | 18481 | 31535 |
| 524288 | 64 | 38334 | 38331 | 114690 | 114697 | 42984 | 38260 |
| 1048576 | 4 | 19803 | 20612 | 114537 | 114742 | 6105 | 7529 |
| 1048576 | 16 | 37983 | 38052 | 114569 | 114715 | 17819 | 27388 |
| 1048576 | 64 | 38297 | 38297 | 114742 | 114703 | 42440 | 38215 |

## Ceph 集群读写测试

由于开始测试时，第一次获得的数据波幅较大，因此对其做了 2 次测试。

V3 版本中，我们加入了 iozone 的 I 参数，以获得更真实的数据。

# Ceph 集群的测试 log-Cached

root@cephfsclient:/home/romi# cat test-write-read.log

```
        Iozone: Performance Test of File I/O
                Version $Revision: 3.397 $
                Compiled for 64 bit mode.
                Build: linux-AMD64

        Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                     Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                     Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                     Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                     Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                     Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                     Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer.
                     Ben England.

        Run began: Mon Mar 10 17:14:18 2014

        Excel chart generation enabled
        Using minimum file size of 131072 kilobytes.
        Using maximum file size of 1048576 kilobytes.
        Record Size 4 KB
        Record Size 16 KB
        Record Size 64 KB
        Command line used: iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r
16K -r 64K -f /mnt/mycephfs/iozonetest
        Output is in Kbytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 Kbytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
                                                                    random    random
bkwd   record   stride
              KB  reclen   write rewrite    read    reread    read   write    read
rewrite     read   fwrite frewrite   fread  freread
          131072       4  224863 1037553  1672750  1710493 1470667  983403
          131072      16   75817 1251200  2986572  3008852 2809566 1276759
          131072      64  137831 1309569  3824446  3840476 3709182 1362309
          262144       4  171053  911592  1668728  1722330 1481718  896292
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 262144 | 16 | 175537 | 1089063 | 2944777 | 2962375 | 2836689 | 1130579 |
| 262144 | 64 | 298722 | 1069451 | 111086 | 3315968 | 3606483 | 1219160 |
| 524288 | 4 | 39343 | 58135 | 1685764 | 1724284 | 1470307 | 4253 |
| 524288 | 16 | 47743 | 47480 | 2898601 | 2960256 | 2692925 | 14475 |
| 524288 | 64 | 61582 | 45115 | 3810565 | 3981380 | 3828597 | 38506 |
| 1048576 | 4 | 33910 | 34316 | 103586 | 1478329 | 1425439 | 1332 |
| 1048576 | 16 | 34250 | 34825 | 2764226 | 3040973 | 2774605 | 5571 |
| 1048576 | 64 | 31299 | 36823 | 3444334 | 3935341 | 3686432 | 15204 |

iozone test complete.
Excel output is below:

"Writer report"

| | "4" | "16" | "64" |
|---|---|---|---|
| "131072" | 224863 | 75817 | 137831 |
| "262144" | 171053 | 175537 | 298722 |
| "524288" | 39343 | 47743 | 61582 |
| "1048576" | 33910 | 34250 | 31299 |

"Re-writer report"

| | "4" | "16" | "64" |
|---|---|---|---|
| "131072" | 1037553 | 1251200 | 1309569 |
| "262144" | 911592 | 1089063 | 1069451 |
| "524288" | 58135 | 47480 | 45115 |
| "1048576" | 34316 | 34825 | 36823 |

"Reader report"

| | "4" | "16" | "64" |
|---|---|---|---|
| "131072" | 1672750 | 2986572 | 3824446 |
| "262144" | 1668728 | 2944777 | 111086 |
| "524288" | 1685764 | 2898601 | 3810565 |
| "1048576" | 103586 | 2764226 | 3444334 |

"Re-Reader report"

| | "4" | "16" | "64" |
|---|---|---|---|
| "131072" | 1710493 | 3008852 | 3840476 |
| "262144" | 1722330 | 2962375 | 3315968 |
| "524288" | 1724284 | 2960256 | 3981380 |
| "1048576" | 1478329 | 3040973 | 3935341 |

"Random read report"

| | "4" | "16" | "64" |
|---|---|---|---|

```
"131072"    1470667   2809566   3709182
"262144"    1481718   2836689   3606483
"524288"    1470307   2692925   3828597
"1048576"    1425439   2774605   3686432

"Random write report"
          "4"   "16"   "64"
"131072"    983403   1276759   1362309
"262144"    896292   1130579   1219160
"524288"    4253   14475   38506
"1048576"    1332   5571   15204
```

## Ceph 集群的二次测试 log-Cached

我们发现第一次 Ceph 的写数据波动较大，因此再测一次，以求有益于参考。

root@cephfsclient:/home/romi# cat test-write-read-v2.log

```
        Iozone: Performance Test of File I/O
                Version $Revision: 3.397 $
                Compiled for 64 bit mode.
                Build: linux-AMD64

        Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                        Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                        Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                        Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                        Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                        Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                        Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer.
                        Ben England.

        Run began: Mon Mar 10 18:07:19 2014

        Excel chart generation enabled
        Using minimum file size of 131072 kilobytes.
        Using maximum file size of 1048576 kilobytes.
        Record Size 4 KB
```

Record Size 16 KB
Record Size 64 KB
Command line used: iozone -Rb ./test-write-read-v2.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -f /mnt/mycephfs/iozonetest
Output is in Kbytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 Kbytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.

| KB | reclen | write | rewrite | read | reread | random read | random write | bkwd read | record rewrite | stride read | fwrite | frewrite | fread | freread |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 131072 | 4 | 83904 | 1022994 | 1673412 | 1709515 | 1503240 | 985948 | | | | | | | |
| 131072 | 16 | 38091 | 1228485 | 2949811 | 2971334 | 2781252 | 1253748 | | | | | | | |
| 131072 | 64 | 56123 | 1254781 | 3679196 | 3915059 | 3792575 | 1377068 | | | | | | | |
| 262144 | 4 | 65393 | 1004044 | 1750043 | 1734421 | 1521289 | 974527 | | | | | | | |
| 262144 | 16 | 255204 | 1195993 | 3002138 | 3000115 | 2794590 | 1195797 | | | | | | | |
| 262144 | 64 | 225729 | 1313668 | 3896368 | 3979789 | 3848499 | 1348068 | | | | | | | |
| 524288 | 4 | 45920 | 65034 | 1678758 | 1714558 | 1458024 | 5223 | | | | | | | |
| 524288 | 16 | 50058 | 48519 | 3099268 | 3113715 | 2875664 | 20228 | | | | | | | |
| 524288 | 64 | 53816 | 56103 | 3844259 | 3865342 | 3527921 | 37278 | | | | | | | |
| 1048576 | 4 | 40288 | 37899 | 110866 | 1534773 | 1506518 | 1471 | | | | | | | |
| 1048576 | 16 | 36067 | 34935 | 109116 | 2665095 | 2713798 | 6355 | | | | | | | |
| 1048576 | 64 | 35454 | 37209 | 3441348 | 3884836 | 4147176 | 18511 | | | | | | | |

iozone test complete.
Excel output is below:

"Writer report"
"4" "16" "64"
"131072" 83904 38091 56123
"262144" 65393 255204 225729
"524288" 45920 50058 53816
"1048576" 40288 36067 35454

"Re-writer report"
"4" "16" "64"
"131072" 1022994 1228485 1254781
"262144" 1004044 1195993 1313668
"524288" 65034 48519 56103
"1048576" 37899 34935 37209

```
"Reader report"
          "4"    "16"    "64"
"131072"     1673412    2949811    3679196
"262144"     1750043    3002138    3896368
"524288"     1678758    3099268    3844259
"1048576"     110866    109116    3441348

"Re-Reader report"
          "4"    "16"    "64"
"131072"     1709515    2971334    3915059
"262144"     1734421    3000115    3979789
"524288"     1714558    3113715    3865342
"1048576"     1534773    2665095    3884836

"Random read report"
          "4"    "16"    "64"
"131072"     1503240    2781252    3792575
"262144"     1521289    2794590    3848499
"524288"     1458024    2875664    3527921
"1048576"     1506518    2713798    4147176

"Random write report"
          "4"    "16"    "64"
"131072"     985948    1253748    1377068
"262144"     974527    1195797    1348068
"524288"     5223    20228    37278
"1048576"     1471    6355    18511
```

## Ceph 集群的测试 log-Non-Cached

root@cephfsclient:/home/romi# **iozone -Rb ./test-write-read.xls -n 128m -g 1G -i 0 -i 1 -i 2 -r 4K -r 16K -r 64K -I -f /mnt/mycephfs/iozonetest |tee ./test-write-read.log &**

Non-Cached 模式无法进行下去，iozone 卡住并中断运行。

## Ceph 集群的测试数据整理-Cached

Ceph 测试的数据(第一次),读写的数据单位为 Kbyte/sec-Cached

| 文件大小 KB | 读写块 KB | 顺序写 Write | 顺序重写 rewrite | 顺序读 read | 顺序重读 reread | 随机读 random read | 随机写 random write |
|---|---|---|---|---|---|---|---|
| 131072 | 4 | 224863 | 1037553 | 1672750 | 1710493 | 1470667 | 983403 |
| 131072 | 16 | 75817 | 1251200 | 2986572 | 3008852 | 2809566 | 1276759 |
| 131072 | 64 | 137831 | 1309569 | 3824446 | 3840476 | 3709182 | 1362309 |
| 262144 | 4 | 171053 | 911592 | 1668728 | 1722330 | 1481718 | 896292 |
| 262144 | 16 | 175537 | 1089063 | 2944777 | 2962375 | 2836689 | 1130579 |
| 262144 | 64 | 298722 | 1069451 | 111086 | 3315968 | 3606483 | 1219160 |
| 524288 | 4 | 39343 | 58135 | 1685764 | 1724284 | 1470307 | 4253 |
| 524288 | 16 | 47743 | 47480 | 2898601 | 2960256 | 2692925 | 14475 |
| 524288 | 64 | 61582 | 45115 | 3810565 | 3981380 | 3828597 | 38506 |
| 1048576 | 4 | 33910 | 34316 | 103586 | 1478329 | 1425439 | 1332 |
| 1048576 | 16 | 34250 | 34825 | 2764226 | 3040973 | 2774605 | 5571 |
| 1048576 | 64 | 31299 | 36823 | 3444334 | 3935341 | 3686432 | 15204 |

Ceph 测试的数据(第二次),读写的数据单位为 Kbyte/sec-Cached

| 文件大小 KB | 读写块 KB | 顺序写 Write | 顺序重写 rewrite | 顺序读 read | 顺序重读 reread | 随机读 random read | 随机写 random write |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 131072 | 4 | 83904 | 1022994 | 1673412 | 1709515 | 1503240 | 985948 |
| 131072 | 16 | 38091 | 1228485 | 2949811 | 2971334 | 2781252 | 1253748 |
| 131072 | 64 | 56123 | 1254781 | 3679196 | 3915059 | 3792575 | 1377068 |
| 262144 | 4 | 65393 | 1004044 | 1750043 | 1734421 | 1521289 | 974527 |
| 262144 | 16 | 255204 | 1195993 | 3002138 | 3000115 | 2794590 | 1195797 |
| 262144 | 64 | 225729 | 1313668 | 3896368 | 3979789 | 3848499 | 1348068 |
| 524288 | 4 | 45920 | 65034 | 1678758 | 1714558 | 1458024 | 5223 |
| 524288 | 16 | 50058 | 48519 | 3099268 | 3113715 | 2875664 | 20228 |
| 524288 | 64 | 53816 | 56103 | 3844259 | 3865342 | 3527921 | 37278 |
| 1048576 | 4 | 40288 | 37899 | 110866 | 1534773 | 1506518 | 1471 |
| 1048576 | 16 | 36067 | 34935 | 109116 | 2665095 | 2713798 | 6355 |
| 1048576 | 64 | 35454 | 37209 | 3441348 | 3884836 | 4147176 | 18511 |

# 大文件-正常操作直接文件复制时间比较

由于 iozone 使用 I 选项时无法对 ceph 客户端挂载的目录进行读写测试，因此，我希望按着最正常的用户操作去感受实际性能。意即分别在 GlusterFS 与 CephFS 测试用的客户端中向挂载的 GlusterFS 及 Ceph 目录中复制文件。这种性能对我们正常操作最实际。

测试命令方法是在客户端执行：time cp file to Gluster/ceph 挂载目录 这样的写或反向读

我分别选 4G,2.5G,1G,200M 左右的文件进行测试，测试均为异步进行，即同一时间只在一个客户端操作。

**关于清除客户端的内存缓存**

另外，为避免客户端内存缓存的影响，特别对读操作的影响，每次操作前若能执行一次清除内存缓存的操作，有可能会好一些，根据此文件经验：
http://wenku.baidu.com/link?url=N2WNW3O2jKyJDJ486hEoL1Cwwu0nwby4yFNUdS-rnaWteoWM5lyT-lrYgRcKhMHskBju-3QAJxhs022vylsTmSuruyuXh1r_p2KXFkGGzEG

sync && echo 3 >/proc/sys/vm/drop_caches && sleep 2 && echo 0 >/proc/sys/vm/drop_caches

但由于操作此后获得的数据更加不太容相信（GlusterFS 读 4G 文件只有 31sec），因此未在以下测试中执行此命令。

# 向 Gluster/Ceph 中写入文件操作记录

复制 4G 文件：

GlusterFS 表现：
[root@gluserclient ~]# ls -l
-rw-r--r--. 1 root root 4239785984 Mar 12 18:02 windowsxp_sp3.img

[root@gluserclient ~]# time cp windowsxp_sp3.img /mnt/sr33

real 5m31.654s
user 0m0.096s
sys 0m10.629s

Ceph 表现：
root@cephfsclient:/home/romi# time cp windowsxp_sp3.img /mnt/mycephfs/

real 2m27.543s
user 0m0.004s
sys 0m8.924s

复制 2.5G 文件：
GlusterFS 表现为：
[root@gluserclient ~]# ls -l
total 2473936
-rw-------. 1 root root 2531524608 Mar 12 17:33 aipu-windows2003-64.qcow2

[root@gluserclient ~]# time cp aipu-windows2003-64.qcow2 /mnt/sr33
cp: overwrite `/mnt/sr33/aipu-windows2003-64.qcow2'? y

real 2m36.842s
user 0m0.039s
sys 0m6.369s

Ceph 的表现为：1m21s
root@cephfsclient:/home/romi# time cp aipu-windows2003-64.qcow2 /mnt/mycephfs/

real 1m21.645s
user 0m0.008s
sys 0m5.356s

复制 1G 文件：

[root@gluserclient ~]# ls -l
total 7546584
-rw-------. 1 root root 954597376 Mar 12 18:27 aipu-centos62-64-mini.qcow2

[root@gluserclient ~]# time cp aipu-centos62-64-mini.qcow2 /mnt/sr33

real 0m34.597s
user 0m0.017s
sys 0m1.918s

root@cephfsclient:/home/romi# time cp aipu-centos62-64-mini.qcow2 /mnt/mycephfs/

real 0m34.378s
user 0m0.004s
sys 0m2.036s

210M 左右的文件复制：

[root@gluserclient ~]# ls -l
-rw-------. 1 root root 213581824 Mar 12 18:37 fedora16-x86_64-openstack-sda.qcow2

[root@gluserclient ~]# time cp fedora16-x86_64-openstack-sda.qcow2 /mnt/sr33

GluserFS 的表现：
real 0m5.543s
user 0m0.003s
sys 0m0.360s

Ceph 的表现：
root@cephfsclient:/home/romi# time cp fedora16-x86_64-openstack-sda.qcow2 /mnt/mycephfs/

real 0m0.892s
user 0m0.000s
sys 0m0.276s

# 从 Gluster/Ceph 中读出文件操作记录

读出 4G 左右文件：

```
[root@gluserclient sr33]# time cp windowsxp_sp3.img /root
cp: overwrite `/root/windowsxp_sp3.img'? y

real    4m11.290s
user    0m0.077s
sys     0m10.563s

root@cephfsclient:/mnt/mycephfs# time cp windowsxp_sp3.img /home/romi

real    1m9.603s
user    0m0.020s
sys     0m8.600s
```

读出 2.5G 左右文件：

```
[root@gluserclient sr33]# time cp aipu-centos62-64-mini.qcow2 /root
cp: overwrite `/root/aipu-centos62-64-mini.qcow2'? y

real    0m12.479s
user    0m0.015s
sys     0m3.005s

root@cephfsclient:/mnt/mycephfs# time cp aipu-centos62-64-mini.qcow2 /home/romi

real    0m9.623s
user    0m0.000s
sys     0m1.892s
```

读出 1G 左右文件

```
[root@gluserclient sr33]# time cp aipu-centos62-64-mini.qcow2 /root
cp: overwrite `/root/aipu-centos62-64-mini.qcow2'? y

real    0m12.619s
user    0m0.012s
sys     0m2.921s
```

root@cephfsclient:/mnt/mycephfs# time cp aipu-centos62-64-mini.qcow2 /home/romi

real 0m10.206s
user 0m0.004s
sys 0m1.708s

读出 210M 左右文件
[root@gluserclient sr33]# time cp fedora16-x86_64-openstack-sda.qcow2 /root
cp: overwrite `/root/fedora16-x86_64-openstack-sda.qcow2'? y

real 0m5.026s
user 0m0.008s
sys 0m0.753s

root@cephfsclient:/mnt/mycephfs# time cp fedora16-x86_64-openstack-sda.qcow2 /home/romi

real 0m2.349s
user 0m0.000s
sys 0m0.484s

## 大文件操作数据整理

我希望按着最正常的用户操作去感受实际性能。意即分别在 GlusterFS 与 CephFS 测试用的客户端中向挂载的 GlusterFS 及 Ceph 目录中复制文件。这种性能对我们正常操作最实际。

测试命令方法是在客户端执行：time cp file to Gluster/ceph 挂载目录 这样的写或反向读

我分别选 4G,2.5G,1G,200M 左右的文件进行测试，测试均为异步进行，即同一时间只在一个客户端操作。

不同大小文件的写耗时：

| 模式 | 文件大小(左右) | GlusterFS(分.秒) | Ceph（分.秒） |
|---|---|---|---|
| 写 | 4G | 5’31” | 2’27” |
| | 2.5G | 2’36” | 1’21” |
| | 1G | 0’34” | 0’34” |
| | 210M | 0’5” | 0’.0.8” |

不同大小文件的读耗时：

| 模式 | 文件大小(左右) | GlusterFS(分.秒) | Ceph（分.秒） |
|---|---|---|---|
| 读 | 4G | 4’11” | 1’9” |
| | 2.5G | 0’12” | 0’9” |
| | 1G | 0’12” | 0’10” |
| | 210M | 0’5” | 0’2” |

# 小文件测试

## Gluster/Ceph 客户端运行在同一台虚拟机上

为消除 centos 及 ubuntu 操作系统的差异对文件读取速率的影响，此次我在同一台机器上（172.16.112.160）同时安装了 gluster 及 ceph 的客户端相关软件。同时，在一台机器上操

作时，为避免内存缓存对文件读取的影响，Glusterfs 操作完成后，我就再次重启机器，再进行 Ceph 客户端对文件的操作。

我们以 40K 为基准(找了一个 38K 左右的 log 文件作为标准-file38k.txt)，生成 1000, 5000，10000 个文件，这样我们看看速率如何。

/mnt/mycephfs 是 ceph 的挂接点
/mnt/sr33 是 gluster 的挂接点

## 生成小文件的脚本

脚本如下（我们能过调整 i 的数值范围而生成不同数量的小文件）：

root@cephfsclient:/home/romi# cat create40k.sh

```
for((i=1;i<=1024;i++)); do
dd if=/root/file38k.txt of=/root/smbfile/hello.$i bs=4k count=10
echo "hello.$i was created"
done
```

我们准备将这些小文件拷贝到 /mnt/sr33 下面的 smbfile 目录中：

## 1024 个小文件的性能对比测试：

Gluster:
root@cephfsclient:/home/romi/smbfile# time cp * /mnt/sr33/smbfile

real 0m27.957s
user 0m0.040s
sys 0m0.312s

重新起动客户端机器后：
Ceph:
root@cephfsclient:/home/romi/smbfile# time cp * /mnt/mycephfs/smbfile

real 0m8.158s

user 0m0.056s
sys 0m0.208s

## 5000 个小文件的性能对比测试：

root@cephfsclient:/home/romi/smbfile# time cp * /mnt/sr33/smbfile

real 3m15.632s
user 0m0.204s
sys 0m1.604s

重新起动机器后：

root@cephfsclient:/home/romi/smbfile# time cp * /mnt/mycephfs/smbfile

real 0m40.302s
user 0m0.296s
sys 0m2.000s

## 10000 个小文件的性能对比测试：

写入：
Gluster:
root@cephfsclient:/home/romi/smbfile# time cp * /mnt/sr33/smbfile/

real 5m21.581s
user 0m0.504s
sys 0m4.008s

重新启动机器：
Ceph
root@cephfsclient:/home/romi/smbfile# time cp * /mnt/mycephfs/smbfile

real 2m8.483s
user 0m0.540s
sys 0m3.744s

浏览目录：
root@cephfsclient:/home/romi# time ls /mnt/sr33/smbfile/*
real 0m34.306s
user 0m0.568s
sys 0m0.512s

重新启动机器：
root@cephfsclient:/home/romi# time ls /mnt/mycephfs/smbfile/*
real 0m0.599s
user 0m0.296s
sys 0m0.072s

读出：

root@cephfsclient:/mnt/sr33/smbfile# time cp * /home/romi/smbfile

real 0m39.511s
user 0m0.368s
sys 0m2.036s

**Ceph**
root@cephfsclient:/mnt/mycephfs/smbfile# time cp * /home/romi/smbfile

real 0m23.461s
user 0m0.216s
sys 0m2.168s

# 小文件测试整理

38K 小文件，不同数量的写、浏览及读耗时：

| 模式 | 文件数量 | GlusterFS(分.秒) | Ceph（分.秒） |
|---|---|---|---|
| 写 | 1024 | 0’27” | 0’8” |
| | 5000 | 3’15” | 0’40” |
| | 10000 | 5’21” | 2’8” |

| | | | |
|---|---|---|---|
| 浏览 | 10000 | 0’34” | 0’0.5” |
| 读 | 10000 | 0’39” | 0’23” |

# 测试小结

## V1,V2 版测试小结

GlusterFS 的写性能数据较低，但顺序读的数据表现较为稳定，即跑满了 1G 网卡带宽；但其余的数据表现不如人意。

Ceph 的测试数据需要进一步研究，其中有一些读的数据超过了 1G byte/sec, 不知 iozone 为什么会给出这样的数据，也不知道是否与 Ceph 的 Rados 机制有关。

但考虑到 iozone 给出的 Ceph 的顺序读最低数据也是在 100M 左右，因此原则上 Ceph 也是可以跑满 1G 的带宽的; 同时，其它各数据表现也比 GlusterFS 要好一些，虽然这些数据看起来有点超出了 1G byte/sec,有点让人不可思议。

另外，由于生产环境的部署中对 GlusterFS 及 Ceph 均会使用调优方法，因此在使用 iozone 时，我并没有做一些特别设置。

综合 Ceph 的二次测试与 GlusterFS 的数据表明, GlusterFS 与 Ceph 相同的 Striped 3+ Replica 3 数据存贮模式下，Ceph 的两次测试的读写性能数据比 GlusterFS 要好，但 GlusterFS 的读写数据很稳定，而 Ceph 的读写数据有些波幅。

小结如下：

1. GlusterFS 数据读写表现比 Ceph 稳定
2. 当客户端实际应用存在缓存的顺序读写情况下：
   a) Ceph 的写速率超过 GlusterFS
   b) 在 256M 文件以下的 4，16，64K 三种情况下 Ceph 仍是线性表现，在实际应用环境中性能超过 GlusterFS。

c) GlusterFS 可以提供与网卡吞吐量相当的读速率，且稳定

3. 当客户端实际应用存在缓存的随机读写情况下，两者难以有较大差别，相对地好象 GlusterFS 要好一些。

网上目前给出的两者对比数据要么是服务器数量达不到集群要求，要么是 GlusterFS 采用简单的 distributed 与 striped 模式(Intel 给出的报告服务器数量太少，同时也没有关于 GlusterFS 的数据存贮模式说明)，没有考虑到数据按三份存放这种中央存贮系统最重要的数据保存模式，因此，部分测试报告说明 GlusterFS 性能比 Ceph 好这种信息。

此份报告只是仅用以参考，因为 Ceph 与 GlusterFS 在各方面差别较大，单纯的比较意义并不是很大。

## V3 版本下文件实操性能

原本计划利用 iozone 的 I 参数再对比一下两者的异同，但无奈 iozone 加上 I 参数后，在 ubuntu 的 ceph 客户端无法正常执行下去，因此无法实现。

但我们利用了实际文件写入及读出的测试，结果表明 ceph 还是好一些。有可能与 Ceph 客户端本身某种机制有关。

同时也有可能与 Centos 与 ubuntu 某些差异导致。

# 联系信息

成都信立讯科技有限公司
成都市科华北路 58 号亚太广场 C 座 711 室，610041
联系电话：13618051964
邮件地址：romizhang1968@gmail.com