[**理解 OpenStack + Ceph （7）: Ceph 的基本操作和常见故障排除方法**](http://www.cnblogs.com/sammyliu/p/5555218.html)

本系列文章会深入研究 Ceph 以及 Ceph 和 OpenStack 的集成：

（1）[安装和部署](http://www.cnblogs.com/sammyliu/p/4804037.html%20)

（2）[Ceph RBD 接口和工具](http://www.cnblogs.com/sammyliu/p/4838139.html%20)

（3）[Ceph 物理和逻辑结构](http://www.cnblogs.com/sammyliu/p/4836014.html%20)

（4）[Ceph 的基础数据结构](http://www.cnblogs.com/sammyliu/p/4843812.html%20)

（5）[Ceph 与 OpenStack 集成的实现](http://www.cnblogs.com/sammyliu/p/4838138.html)

（6）[QEMU-KVM 和 Ceph RBD 的 缓存机制总结](http://www.cnblogs.com/sammyliu/p/5066895.html)

（7）[Ceph 的基本操作和常见故障排除方法](http://www.cnblogs.com/sammyliu/p/5555218.html%20)

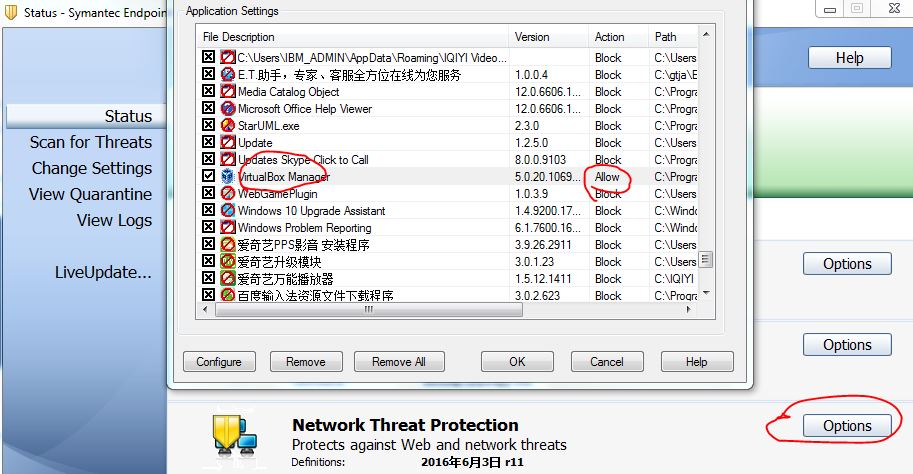
学以致用，本文将介绍 Ceph 集群的一些基本操作和常见的故障排除方法。

**0. 测试环境**

  将由 Virtulbox 管理的两个虚机，分别是 ceph1 和 ceph2，作为 OSD 服务器，其中，ceph1 同时作为 MON 服务器。两个节点上分别有两个虚拟磁盘作为 OSD 存储磁盘，每个大小为 5G；还有一个虚拟磁盘作为 Journal 磁盘，1G，分为两个区，做为数据盘的日志分区。从 ceph1 上使用 ceph-deploy 工具部署。pool 的 size 设置为 2，min\_size 也设置为 2.

  每个机器配置两个网卡，eth0 是 NAT 模式，使用 DHCP 方式获取IP，用于虚机访问外网；eth1 是 host-only 模式，配置静态IP，用于主机访问虚机，以及虚机之间的互访。对Ceph 来说，eth0 用于安装程序，eth1 用于 public 和 cluster network。

 注意在宿主机上要设置防火墙允许 Virtualbox 访问外网。我机器上安装了 Symentac Endpoint Protection，不小心 block 了它的互联网访问，然后虚机也就无法连接外网了，我在这里花了不少时间折腾：



**1. 修改 OSD CRUSH weight**

**1.1 问题描述**

部署完成后，集群处于 PG Degraded 状态，经查 ceph health detail，发现 PG 的 acting OSD 只有 [0]，而不是两个。查 osd tree，osd 日志等，看不出明显问题。

**1.2 原因分析**

  从文章 <http://serverfault.com/questions/680492/after-initial-deployment-ceph-cluster-stays-in-activedegraded-state> 受到启发，我的 Ceph 集群的 OSD 的 weight 都是 0！！

[复制代码](javascript:void(0);)

root@ceph1:/etc/ceph# ceph osd tree

# id weight type name up/down reweight

-1 0 root default

-2 0 host ceph1

0 0 osd.0 up 1

2 0 osd.2 up 1

-3 0 host ceph2

1 0 osd.1 up 1

3 0 osd.3 up 1

[复制代码](javascript:void(0);)

  我们从上面 ceph osd tree 的结果里面可以看到这里有两个weight：weight 和 reweight。[这篇文章](http://hustcat.github.io/difference_between_osd_weight_and_osd_crush_weight/) 有详细的分析。简单来说：

* weight：即 osd crush weight，表示设备(device) 容量的相对值，比如如果1TB对应1.00，那么 500MB 对应 0.50。bucket weight 是所有 item weight 之和，item weight 的变化会影响 bucket weight 的变化，也就是 osd.X 会影响host。 对于 straw bucket，如果 item weight 为0，则 item straw 也为0，当CRUSH 算法在 bucket 选择 item 时，也就不太可能选中该 item。
* reweight：取值为0~1。osd reweight 并不会影响 host。当 osd 被踢出集群（out）时，osd weight 被设置0，加入集群时，设置为1。它会参与 CRUSH 创建 PG 的过程。CRUSH在选择 OSD 时，如果发现 weight 为0，就跳过该 OSD。

因此，问题的症结就在于 osd crush weight 为0。至于为什么会这样，以及该值对 PG 分配的影响，有待进一步查明。

**1.3 解决办法：修改 osd crush weight**

ceph osd crush reweight osd.0 1

ceph osd crush reweight osd.1 1

ceph osd crush reweight osd.2 1

ceph osd crush reweight osd.3 1

修改后，集群就回到了 HEALTH\_OK 状态。

注意：修改 OSD 的 crush weight 会带来部分 PG 之间的数据移动，这可能会影响集群的性能，因此在生产环境中使用要小心。你可以参考 [这篇文章](https://ceph.com/planet/ceph-osd-reweight/) 来看数据移动的情况。

**2. 修改 CRUSH tunables（可调参数）**

**2.1 问题描述**

将 osd.1 设置为 out 后，集群并没有开始做 recovery，部分 PG 保持在 remapped 状态：

[复制代码](javascript:void(0);)

root@ceph1:~# ceph -s

cluster 5ccdcb2d-961d-4dcb-a9ed-e8034c56cf71

health HEALTH\_WARN 88 pgs stuck unclean

monmap e2: 1 mons at {ceph1=192.168.56.102:6789/0}, election epoch 1, quorum 0 ceph1

osdmap e71: 4 osds: 4 up, 3 in

pgmap v442: 256 pgs, 4 pools, 285 MB data, 8 objects

690 MB used, 14636 MB / 15326 MB avail

88 active+remapped

168 active+clean

[复制代码](javascript:void(0);)

**2.2 原因分析**

（1）. 查看 ceph health detail

root@ceph1:~# ceph health detail

HEALTH\_WARN 88 pgs stuck unclean

pg 1.23 is stuck unclean for 337.342290, current state active+remapped, last acting [0,1]

pg 0.1f is stuck unclean for 336.838743, current state active+remapped, last acting [0,1]

pg 1.1f is stuck unclean for 337.355851, current state active+remapped, last acting [0,1]

Remapped（重映射）：当 PG 的 acting set 变化后，数据将会从旧 acting set 迁移到新 action set。新主 OSD 需要过一段时间后才能提供服务。因此，它会让老的主 OSD 继续提供服务，直到 PG 迁移完成。数据迁移完成后，PG map 将使用新 acting set 中的主OSD。

以 PG 为例，比较在 osd.1 out 前后的 PG map：

state state\_stamp v reported up up\_primary acting acting\_primary

active+clean 2016-06-03 00:31:44.220896 0'0 57:74 [0,1] 0 [0,1] 0 #osd.1 out 之前

active+remapped 2016-06-03 00:47:12.703537 0'0 71:109 [0] 0 [0,1] 0 #osd.1 out 之后

**2.3 解决办法**

**2.3.1 之一：将 crush tunables 设置为 optimal**

（2）从[这篇文章](http://bbs.ceph.org.cn/question/125)中获得线索，这可能和 crush tunables 有关系。它的默认值应该是 legacy，运行下面的命令将其修改为 optimal 后，集群状态回到正常。

ceph osd crush tunables optimal

（3）继续找原因，Red Hat [这篇文章](https://access.redhat.com/documentation/en/red-hat-ceph-storage/1.3/storage-strategies/chapter-9-crush-tunables) 给出了一些线索。

  在新版本的Ceph 集群中使用 legacy 值可能会有一些问题，包括：

* 当叶子bucket（往往是 host）所拥有的设备数目很小时，一些 PG 被映射到的 OSD 数目少于存储池的size。这在 host 节点的 OSD 数目为 1-3 时较为常见。
* 大型集群中，小部分的 PG 被映射到的 OSD 数目小于规定的数目。这在 CRUSH 层级结构中好几层（比如 row，rack，host，osd 等）时比较常见。
* 当一些 OSD 被标记为 out 时，重新分布的数据会更多地在附近的 OSD 上而不是整个层级结构中。

  而第一种情况正是我的测试集群所遇到的情况，每个 host 拥有的 OSD 数目在3个以内，然后部分 PG 所在的 OSD 数目较 replica 少一些。

  关于 CRUSH tunables 的详细信息，请阅读有关文档。请注意在生产环境操作时有一定的风险。

**2.3.2 之二：将 OSD 的 reweight 修改为 0 而不是使用 out 命令**

    Ceph 官方的[这篇文章](http://docs.ceph.com/docs/jewel/rados/operations/add-or-rm-osds/) 给出了另一个思路。它认为在主机数目很小的集群中，当一个 OSD 被 out 后，部分 PG 限于 active+remapped 状态是经常出现的。解决办法是先运行 ceph osd in {osd-num} 将集群状态恢复到初始状态，然后运行 ceph osd crush reweight osd.{osd-num} 0 来将这个 osd 的 crush weight 修改为 0，然后集群会开始数据迁移。对小集群来说，reweight 命令甚至更好些。

   当集群中 PG 限于 active + remapped 状态时，可以通过 reweight 命令来使得集群恢复正常。当往集群中新加入 OSD 时，为了减少数据移动对集群性能的影响，Ceph 官方建议逐渐地增加 OSD 的 crush weight，比如起始值为0，先设置为 0.2，等数据迁移结束，再设置为 0.4，依此类推，逐渐增加为 0.6,0.8 和 1 甚至更高。在要停用一个 OSD 时，建议采用相反操作，逐渐减少 OSD 的 crush weight 直至 0.

**3. 修改 CRUSH ruleset**

**3.1 问题描述**

继续将跟 osd.1 在同意个host 上的 osd.3 out，看看 Ceph 集群能不能继续恢复。Ceph 集群中部分 PG 再次进入 remapped 状态：

[复制代码](javascript:void(0);)

root@ceph1:~# ceph -s

cluster 5ccdcb2d-961d-4dcb-a9ed-e8034c56cf71

health HEALTH\_WARN 256 pgs stuck unclean

monmap e2: 1 mons at {ceph1=192.168.56.102:6789/0}, election epoch 1, quorum 0 ceph1

osdmap e77: 4 osds: 4 up, 2 in

pgmap v480: 256 pgs, 4 pools, 285 MB data, 8 objects

625 MB used, 9592 MB / 10217 MB avail

256 active+remapped

[复制代码](javascript:void(0);)

运行 ceph pg 1.0 query 查看 PG 1.0 的状态：

[复制代码](javascript:void(0);)

"recovery\_state": [

{ "name": "Started\/Primary\/Active",

"enter\_time": "2016-06-03 01:31:22.045434",

"might\_have\_unfound": [],

"recovery\_progress": { "backfill\_targets": [],

"waiting\_on\_backfill": [],

"last\_backfill\_started": "0\/\/0\/\/-1",

"backfill\_info": { "begin": "0\/\/0\/\/-1",

"end": "0\/\/0\/\/-1",

"objects": []},

"peer\_backfill\_info": [],

"backfills\_in\_flight": [],

"recovering": [],

"pg\_backend": { "pull\_from\_peer": [],

"pushing": []}},

"scrub": { "scrubber.epoch\_start": "0",

"scrubber.active": 0,

"scrubber.block\_writes": 0,

"scrubber.finalizing": 0,

"scrubber.waiting\_on": 0,

"scrubber.waiting\_on\_whom": []}},

{ "name": "Started",

"enter\_time": "2016-06-03 01:31:20.976290"}],

[复制代码](javascript:void(0);)

可见它已经开始 recovery 了，但是没完成。

**3.2 原因分析**

PG 的分布和 CRUSH ruleset 有关。我的集群当前只有一个默认的 ruleset：

[复制代码](javascript:void(0);)

root@ceph1:~# ceph osd crush rule dump

[

{ "rule\_id": 0,

"rule\_name": "replicated\_ruleset",

"ruleset": 0,

"type": 1,

"min\_size": 1,

"max\_size": 10,

"steps": [

{ "op": "take",

"item": -1,

"item\_name": "default"},

{ "op": "chooseleaf\_firstn",

"num": 0,

"type": "host"},

{ "op": "emit"}]}]

[复制代码](javascript:void(0);)

注意其 type 为 “host”，也就是说 CRUSH 不会为一个 PG 选择在同一个 host 上的两个 OSD。而我的环境中，目前只有 ceph1 上的两个 OSD 是in 的，因此，CRUSH 无法为所有的 PG 重新选择一个新的 OSD 来替代 osd.3.

**3.3 解决办法**

按照以下步骤，将 CRUSH ruleset 的 type 由 “host” 修改为 “osd”，使得 CRUSH 为 PG 选择 OSD 时不再局限于不同的 host。

root@ceph1:~# ceph osd getcrushmap -o crushmap\_compiled\_file

got crush map from osdmap epoch 77

root@ceph1:~# crushtool -d crushmap\_compiled\_file -o crushmap\_decompiled\_file

root@ceph1:~# vi crushmap\_decompiled\_file

[复制代码](javascript:void(0);)

rule replicated\_ruleset {

ruleset 0

type replicated

min\_size 1

max\_size 10

step take default

step chooseleaf firstn 0 type osd #将 type 由 “host” 修改为 “osd”

step emit

}

[复制代码](javascript:void(0);)

root@ceph1:~# crushtool -c crushmap\_decompiled\_file -o newcrushmap

root@ceph1:~# ceph osd setcrushmap -i newcrushmap

set crush map

以上命令执行完毕后，可以看到 recovery 过程继续进行，一段时间后，集群恢复 OK 状态。

[复制代码](javascript:void(0);)

root@ceph1:~# ceph -s

cluster 5ccdcb2d-961d-4dcb-a9ed-e8034c56cf71

health HEALTH\_WARN 256 pgs stuck unclean

monmap e2: 1 mons at {ceph1=192.168.56.102:6789/0}, election epoch 1, quorum 0 ceph1

osdmap e80: 4 osds: 4 up, 2 in

pgmap v493: 256 pgs, 4 pools, 285 MB data, 8 objects

552 MB used, 9665 MB / 10217 MB avail

256 active+remapped

root@ceph1:~# ceph -s

cluster 5ccdcb2d-961d-4dcb-a9ed-e8034c56cf71

health HEALTH\_WARN 137 pgs stuck unclean

monmap e2: 1 mons at {ceph1=192.168.56.102:6789/0}, election epoch 1, quorum 0 ceph1

osdmap e80: 4 osds: 4 up, 2 in

pgmap v494: 256 pgs, 4 pools, 285 MB data, 8 objects

677 MB used, 9540 MB / 10217 MB avail

137 active+remapped

119 active+clean

recovery io 34977 B/s, 0 objects/s

root@ceph1:~# ceph -s

cluster 5ccdcb2d-961d-4dcb-a9ed-e8034c56cf71

health HEALTH\_OK

monmap e2: 1 mons at {ceph1=192.168.56.102:6789/0}, election epoch 1, quorum 0 ceph1

osdmap e80: 4 osds: 4 up, 2 in

pgmap v495: 256 pgs, 4 pools, 285 MB data, 8 objects

679 MB used, 9538 MB / 10217 MB avail

256 active+clean

recovery io 18499 kB/s, 0 objects/s

[复制代码](javascript:void(0);)

**4. 将一个 OSD 移出集群**

（1）将该 osd 设置为 out

root@ceph1:/home/s1# ceph osd out osd.1

marked out osd.1.

（2）集群做 recovery

[复制代码](javascript:void(0);)

2016-06-03 01:54:21.596632 mon.0 [INF] osdmap e90: 4 osds: 4 up, 3 in

2016-06-03 01:54:21.608675 mon.0 [INF] pgmap v565: 256 pgs: 256 active+clean; 1422 MB data, 2833 MB used, 12493 MB / 15326 MB avail

2016-06-03 01:54:26.352909 mon.0 [INF] pgmap v566: 256 pgs: 1 active, 255 active+clean; 1422 MB data, 2979 MB used, 12347 MB / 15326 MB avail; 2/40 objects degraded (5.000%); 51033 B/s, 0 objects/s recovering

2016-06-03 01:54:28.624334 mon.0 [INF] pgmap v567: 256 pgs: 4 active, 252 active+clean; 1422 MB data, 3427 MB used, 11899 MB / 15326 MB avail; 8/40 objects degraded (20.000%); 51053 B/s, 0 objects/s recovering

2016-06-03 01:54:31.320973 mon.0 [INF] pgmap v568: 256 pgs: 3 active, 253 active+clean; 1422 MB data, 3539 MB used, 11787 MB / 15326 MB avail; 6/40 objects degraded (15.000%); 19414 kB/s, 0 objects/s recovering

2016-06-03 01:54:32.323443 mon.0 [INF] pgmap v569: 256 pgs: 256 active+clean; 1422 MB data, 3730 MB used, 11595 MB / 15326 MB avail; 77801 kB/s, 0 objects/s recovering

^[[A2016-06-03 01:56:10.949077 mon.0 [INF] pgmap v570: 256 pgs: 256 active+clean; 1422 MB data, 3730 MB used, 11595 MB / 15326 MB avail

[复制代码](javascript:void(0);)

（3）完成后，该 osd 的状态还是 up，表示它的服务还在运行。现在将其服务停掉。

root@ceph1:/home/s1# ssh ceph2 service ceph stop osd.1

/etc/init.d/ceph: osd.1 not found (/etc/ceph/ceph.conf defines , /var/lib/ceph defines )

该命令出错，需要将 osd.1 加入 ceph.conf 中。在 ceph1 上的 ceph.conf 中添加：

[复制代码](javascript:void(0);)

[osd]

[osd.1]

host = ceph2

[osd.2]

host = ceph1

[osd.3]

host = ceph2

[osd.0]

host = ceph1

[复制代码](javascript:void(0);)

然后运行 ceph-deploy --overwrite-conf config push ceph2 将它拷贝到 ceph2 上。重启所有的 osd 服务。诡异的事情出现了：

[复制代码](javascript:void(0);)

root@ceph1:/etc/ceph# ceph osd tree

# id weight type name up/down reweight

-1 4 root default

-2 4 host ceph1

0 1 osd.0 up 1

2 1 osd.2 up 1

1 1 osd.1 up 0

3 1 osd.3 up 1

-3 0 host ceph2

[复制代码](javascript:void(0);)

osd.1 和 osd.3 跑到了 ceph1 节点上！查看 start 命令，它将 curshmap 中的 osd.1 的 host 修改为了 ceph2：

root@ceph1:/etc/ceph# /etc/init.d/ceph -a start osd

=== osd.1 ===

df: â€˜/var/lib/ceph/osd/ceph-1/.â€™: No such file or directory

create-or-move updating item name 'osd.1' weight 1 at location {host=ceph1,root=default} to crush map

Starting Ceph osd.1 on ceph2...

starting osd.1 at :/0 osd\_data /var/lib/ceph/osd/ceph-1 /var/lib/ceph/osd/ceph-1/journal

从 [这篇文章](http://tracker.ceph.com/issues/6227) 可以看出，这其实是Ceph的一个 bug：make osd crush placement on startup handle multiple trees (e.g., ssd + sas)。该bug 在 [OSD location reset after restart](http://lists.ceph.com/pipermail/ceph-users-ceph.com/2014-April/038835.html) 中也有讨论。目前 Ceph 没有机制可以确保 CRUSH map 结构不变，最简单的办法是在 ceph.conf 中 [OSD] 部分设置 osd crush update on start  = false。

尝试手工挪动 osd.1 和 osd.3：

root@ceph1:/etc/ceph# ceph osd crush remove osd.1

removed item id 1 name 'osd.1' from crush map

root@ceph1:/etc/ceph# ceph osd crush remove osd.3

removed item id 3 name 'osd.3' from crush map

[复制代码](javascript:void(0);)

root@ceph1:/etc/ceph# ceph osd tree

# id weight type name up/down reweight

-1 2 root default

-2 2 host ceph1

0 1 osd.0 up 1

2 1 osd.2 up 1

-3 0 host ceph2

1 0 osd.1 up 0

3 0 osd.3 up 1

[复制代码](javascript:void(0);)

root@ceph1:/etc/ceph# ceph osd crush set 1 1 root=default host=ceph2

Error ENOENT: unable to set item id 1 name 'osd.1' weight 1 at location {host=ceph2,root=default}: does not exist

该错误的原因待查。索性直接修改 crush map，然后正确的结果就回来了：

[复制代码](javascript:void(0);)

root@ceph1:/etc/ceph# ceph osd tree

# id weight type name up/down reweight

-1 2 root default

-2 2 host ceph1

0 1 osd.0 up 1

2 1 osd.2 up 1

-3 0 host ceph2

1 1 osd.1 up 0

3 1 osd.3 up 1

[复制代码](javascript:void(0);)

继续运行命令 ssh ceph2 /etc/init.d/ceph stop osd.1 去停止 osd.1 的服务，但是无法停止。据说是因为用 ceph-deploy 部署的 OSD 的服务都没法停止。只能想办法把进程杀掉了。

然后继续执行：

root@ceph1:/etc/ceph# ceph osd crush remove osd.1

removed item id 1 name 'osd.1' from crush map

root@ceph1:/etc/ceph# ceph auth del osd.1

updated  
root@ceph1:/etc/init# ceph osd rm osd.1  
removed osd.1

此时，osd tree 中再也没有 osd.1 了：

[复制代码](javascript:void(0);)

root@ceph1:/etc/ceph# ceph osd tree

# id weight type name up/down reweight

-1 3 root default

-2 2 host ceph1

0 1 osd.0 up 1

2 1 osd.2 up 1

-3 1 host ceph2

3 1 osd.3 up 1

[复制代码](javascript:void(0);)

**5. 将一个 OSD 加入集群**

 （1）将 /dev/sdb1 分区删除

（2）清理磁盘：ceph-deploy disk zap ceph2:/dev/sdb

（3）创建 OSD：ceph-deploy osd create ceph2:sdb:/dev/sdd1

结果OSD就回来了：

[复制代码](javascript:void(0);)

root@ceph1:~# ceph-deploy osd create ceph2:sdb:/dev/sdd1c^C

root@ceph1:~# ceph osd tree

# id weight type name up/down reweight

-1 2 root default

-2 2 host ceph1

0 1 osd.0 up 1

2 1 osd.2 up 1

-3 0 host ceph2

4 0 osd.4 up 1

1 0 osd.1 up 1

[复制代码](javascript:void(0);)

其实将上面第四步和第五步合并在一起，就是替换一个故障磁盘的过程。

**6. 在特定 OSD 上创建存储池**

    我们假设 osd.0 和 osd.2 的磁盘是 SSD 磁盘，osd.1 和 osd.4 的磁盘是 SATA 磁盘。我们将创建两个pool：pool-ssd 和 pool-sata，并确保 pool-ssd 中的对象都保存在 osd.0 和 osd.2 上，pool-sata 中的对象都保存在 osd.1 和 osd.4 上。

（1）修改 CRUSH map

root@ceph1:~# ceph osd getcrushmap -o crushmapdump

got crush map from osdmap epoch 124

root@ceph1:~# crushtool -d crushmapdump -o crushmapdump-decompiled

root@ceph1:~# vi crushmapdump-decompiled  
root@ceph1:~# crushtool -c crushmapdump-decompiled -o crushmapdump-compiled  
root@ceph1:~# ceph osd setcrushmap -i crushmapdump-compiled

在 crushmapdump-decompiled 文件中添加如下内容：

[复制代码](javascript:void(0);)

root ssd {

id -5

alg straw

hash 0

item osd.0 weight 1

item osd.2 weight 1

}

root sata {

id -6

alg straw

hash 0

item osd.1 weight 1

item osd.4 weight 1

}

# rules

...

rule ssd-pool {

ruleset 1

type replicated

min\_size 1

max\_size 10

step take ssd

step chooseleaf firstn 0 type osd

step emit

}

rule sata-pool {

ruleset 2

type replicated

min\_size 1

max\_size 10

step take sata

step chooseleaf firstn 0 type osd

step emit

}

[复制代码](javascript:void(0);)

（2） ceph osd tree 如下：

[复制代码](javascript:void(0);)

root@ceph1:~# ceph osd tree

# id weight type name up/down reweight

-6 2 root sata

1 1 osd.1 up 1

4 1 osd.4 up 1

-5 2 root ssd

0 1 osd.0 up 1

2 1 osd.2 up 1

-1 2 root default

-2 2 host ceph1

0 1 osd.0 up 1

2 1 osd.2 up 1

-3 0 host ceph2

4 0 osd.4 up 1

1 0 osd.1 up 1

[复制代码](javascript:void(0);)

（3）创建 ssd-pool，其默认的 ruleset 为 0：

root@ceph1:~# ceph osd pool create ssd-pool 8 8

pool 'ssd-pool' created

root@ceph1:~# ceph osd dump | grep -i ssd

pool 4 'ssd-pool' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 8 pgp\_num 8 last\_change 126 flags hashpspool stripe\_width 0

（4）修改 ssd-pool 的 ruleset 为 ssd-pool 其id 为 1：

root@ceph1:~# ceph osd pool set ssd-pool crush\_ruleset 1

set pool 4 crush\_ruleset to 1

root@ceph1:~# ceph osd dump | grep -i ssd

pool 4 'ssd-pool' replicated size 2 min\_size 1 crush\_ruleset 1 object\_hash rjenkins pg\_num 8 pgp\_num 8 last\_change 128 flags hashpspool stripe\_width 0

（5）类似地创建 sata-pool 并设置其 cursh ruleset 为 sata-pool 其id 为 2：

root@ceph1:~# ceph osd pool create sata-pool 8 8

pool 'sata-pool' created

root@ceph1:~# ceph osd pool set sata-pool crush\_ruleset 2

set pool 5 crush\_ruleset to 2

root@ceph1:~# ceph osd dump | grep -i sata

pool 5 'sata-pool' replicated size 2 min\_size 1 crush\_ruleset 2 object\_hash rjenkins pg\_num 8 pgp\_num 8 last\_change 131 flags hashpspool stripe\_width 0

（6）分别放一个文件进这两个pool：

root@ceph1:/home/s1# rados -p ssd-pool put root-id\_rsa root-id\_rsa

root@ceph1:/home/s1# rados -p sata-pool put root-id\_rsa root-id\_rsa

root@ceph1:/home/s1# rados -p ssd-pool ls

root-id\_rsa

root@ceph1:/home/s1# rados -p sata-pool ls

root-id\_rsa

（7）查看对象所在的 OSD

root@ceph1:/home/s1# ceph osd map ssd-pool root-id\_rsa

osdmap e132 pool 'ssd-pool' (4) object 'root-id\_rsa' -> pg 4.38e001ef (4.7) -> up ([2,0], p2) acting ([2,0], p2)

root@ceph1:/home/s1# ceph osd map sata-pool root-id\_rsa

osdmap e132 pool 'sata-pool' (5) object 'root-id\_rsa' -> pg 5.38e001ef (5.7) -> up ([4,1], p4) acting ([4,1], p4)

可见，两个pool各自在ssd 和 sata 磁盘上。

**7. rbd map 失败：Input/output error**

**7.1 问题描述**

   先准备好机器，然后在Ceph集群管理节点上运行 ceph-deploy install client 和 ceph-deploy admin client 来安装和配置该节点。详细过程可以参考 [Oracle 这篇文章](http://docs.oracle.com/cd/E37670_01/E66514/html/ceph-ngv_vd1_dt.html)。 注意第二个命令需要在 /etc/ceph 目录下执行，否则，拷贝到 client 节点上的 ceph.conf 会不正确，运行 ceph -s 也会碰到错误 0 librados: client.admin authentication error (1) Operation not permitted。

  然后执行下面的命令来创建一个卷并mount给该客户端：

[复制代码](javascript:void(0);)

root@client:~# rbd create -p pool1 bd1 --size 100

root@client:~# rbd info --image bd1 -p pool1

rbd image 'bd1':

size 102400 kB in 25 objects

order 22 (4096 kB objects)

block\_name\_prefix: rb.0.383a.2ae8944a

format: 1

root@client:~# rbd map -p pool1 bd1

rbd: add failed: (5) Input/output error

[复制代码](javascript:void(0);)

**7.2 问题解决和原因分析**

从 [这篇文章](http://lists.ceph.com/pipermail/ceph-users-ceph.com/2014-July/041205.html) 中获得线索，需要修改 crushmap 中的

root@client:/home/s1# ceph osd getcrushmap -o crush

got crush map from osdmap epoch 138

root@client:/home/s1# crushtool -i crush --set-chooseleaf\_vary\_r 0 -o crush.newroot@client:/home/s1# ceph osd setcrushmap -i crush.new

set crush map

root@client:/home/s1# rbd map -p pool1 bd1

初步看起来，这个问题和 linux 内核版本（我的内核版本是 1.13）以及 crush tunables 有关。在 3.15 之前的版本中，chooseleaf\_vary\_r 的值必须为0。根本原因需要进一步研究。

**8. 修复 stale 状态 PG**

**8.1 问题描述**

[复制代码](javascript:void(0);)

root@client:/home/s1# ceph -s

cluster 5ccdcb2d-961d-4dcb-a9ed-e8034c56cf71

health HEALTH\_WARN 8 pgs stale; 8 pgs stuck stale

monmap e2: 1 mons at {ceph1=192.168.56.102:6789/0}, election epoch 1, quorum 0 ceph1

osdmap e182: 4 osds: 4 up, 4 in

pgmap v2569: 336 pgs, 8 pools, 5763 MB data, 1171 objects

13763 MB used, 6672 MB / 20435 MB avail

8 stale+active+clean

328 active+clean

[复制代码](javascript:void(0);)

学习一下 stale 状态的概念：PG 的状态没有被 ceph-osd 上报给 MON，这意味着存放该 PG 的所有 OSD 都是down 的。MON 将主 OSD down 掉了的 PG 的状态状态设置为 stale。

**8.2 问题定位**

（1）.找出哪些 PG 处于 stale 状态

root@ceph1:~# ceph pg dump | grep stale

dumped all in format plain

5.2 0 0 0 0 0 0 0 stale+active+clean 2016-06-03 03:50:07.049571 0'0 132:6 [4,1] 4 [4,1] 4 0'0 2016-06-03 03:49:50.301465 0'0 2016-06-03 03:49:50.301465

5.3 0 0 0 0 0 0 0 stale+active+clean 2016-06-03 03:50:07.034290 0'0 132:6 [4,1] 4 [4,1] 4 0'0 2016-06-03 03:49:50.302140 0'0 2016-06-03 03:49:50.302140

但是现在 OSD Tree 中不再有 osd.1 和 osd.4，因为这两个 OSD 的盘之前丢了！后来重做成了 osd.3 和 osd.5。这和上面的 stale 状态的概念吻合。重做的时候 OSD磁盘的分区被删除过，不知道数据是否还在。看来，丢失数据也是很容易的事情啊。看来，下次看到某个 OSD down 掉了时，首先应该做的是要将它重新 up 起来。

（2）修复不行

root@ceph1:~# ceph pg repair 5.2

Error EAGAIN: pg 5.2 primary osd.4 not up

（3）只能删除它所属于的 pool 了。注意 PG ID 的前半部分是 pool id，后半部分才是 PG 自己的 id。

root@ceph1:~# ceph osd pool delete sata-pool sata-pool --yes-i-really-really-mean-it

pool 'sata-pool' removed

其实这么做也不对，因为一个 pool 有多个 PG，其中只有部分是 stale 状态，因此在删除之前要看清楚。这一步骤要十分小心！

（4）ceph -s 恢复正常。

**9. 分离 public network 和 cluster network**

**9.1 分离的好处**

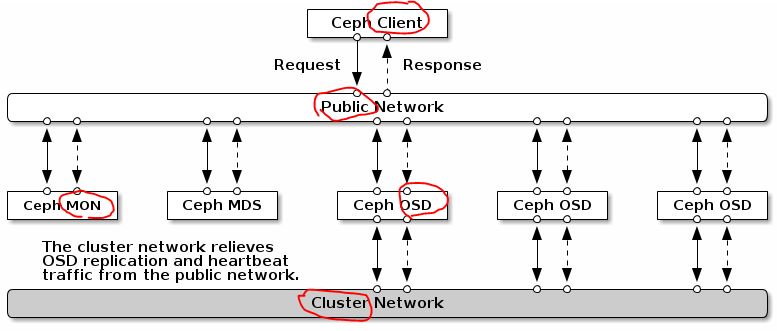
（1）提高性能：消除副本创建、数据恢复和再平衡对 public network 的压力；增强 OSD 心跳网络的可靠性。

（2）安全性：使用一个彻底与外网分离的内部网络作为 cluster network，可以防止比如 DDOS 这样的网络攻击。

更多信息，请参阅 [NETWORK CONFIGURATION REFERENCE](http://docs.ceph.com/docs/jewel/rados/configuration/network-config-ref/)。

**9.2 分离的方法**

（1）配置网络



给每个 OSD 节点增加一块网卡，它的连接方式为 “内部网络”；在虚机内配置静态IP地址，网段为 192.168.1.100/24 （其实用不了这么大的网段）.

（2）在 ceph1 上修改 ceph.conf 文件

[复制代码](javascript:void(0);)

[global]

...

public network = 192.168.56.100/24

cluster network = 192.168.1.100/24

[mon]

[mon.ceph1] # MON 守护进程只在public network 内

host = ceph1

mon addr = 192.168.56.102:6789

[osd]

osd journal size = 500

osd crush update on start = false

[osd.3] #OSD 守护进程同时在 public 和 cluster network 上

host = ceph2

public addr = 192.168.56.103

cluster addr = 192.168.1.103

[osd.0]

host = ceph1

public addr = 192.168.56.102

cluster addr = 192.168.1.102

[osd.5]

host = ceph2

public addr = 192.168.56.103

cluster addr = 192.168.1.103

[osd.2]

host = ceph1

public addr = 192.168.56.102

cluster addr = 192.168.1.102

[复制代码](javascript:void(0);)

（3）将新的 ceph.conf 分发到其它节点上，比如 ceph-deploy --overwrite-conf config push ceph2

（4）重启所有 OSD 和 MON 守护进程

可以在 osd 日志中看到内部网络IP地址被启用了。

**10. 启用客户端 rbd cache**

**10.1 步骤**

编辑在客户端的 ceph.conf 文件，添加下面的配置项

[client]

rbd cache = true

rbd cache writethrough until flush = true

admin socket = /var/run/ceph/$cluster-$type.$id.$pid.$cctid.asok

log file = /var/log/ceph/

启动使用 librbd 的应用，比如 fio + rbd ioengine，然后从 client admin socket 中确认cache 被启用了：

[复制代码](javascript:void(0);)

root@client:/var/run/ceph# ls

ceph-client.admin.23789.140009021853536.asok

root@client:/var/run/ceph# ceph --admin-daemon ceph-client.admin.23789.140009021853536.asok config show | grep rbd\_cache

"rbd\_cache": "true",

"rbd\_cache\_writethrough\_until\_flush": "false",

"rbd\_cache\_size": "33554432",

"rbd\_cache\_max\_dirty": "25165824",

"rbd\_cache\_target\_dirty": "16777216",

"rbd\_cache\_max\_dirty\_age": "1",

"rbd\_cache\_max\_dirty\_object": "0",

"rbd\_cache\_block\_writes\_upfront": "false",

[复制代码](javascript:void(0);)

注意rbd cache 只对 librbd 起作用，对 kernel rbd 不起作用