重点内容

- ceph pool的概念
- ceph PG的概念
- osd和pg数量之间的关系

1 查看pool和pg

SES 7 | 操作和管理指南 | 确定集群状态 (suse.com)

1.1 查看pool

docker exec mon ceph osd lspools

docker exec mon ceph osd lspools

1 .rgw.root
2 default.rgw.control
3 default.rgw.meta
4 default.rgw.log
5 cephfs_data --- 文件系统
6 cephfs_metadata -- 为rgw的bucket信息，比如图片 宽高信息
7 default.rgw.buckets.index -- 为rgw的bucket信息
8 default.rgw.buckets.data 00 -- 是实际存储的数据信息
9 default.rgw.buckets.non-ec

1. .rgw.root：包含realm（领域信息），比如zone和zonegroup
2. default.rgw.control：系统控制池，再有数据更新时，通知其它RGW更新缓存
3. default.rgw.log：存储日志信息，用于记录各种log信息
4. default.rgw.meta：元数据存储池，通过不同的名称空间分别存储不同的rados对象，这些名称空间包括用户uid及其bucket映射信息的名称空间users.uid、用户的密钥名称空间users.keys、用户的email名称空间users.email、用户的subuser的名称空间users.swift，以及bucket的名称空间root等
5. default.rgw.buckets.index：存放bucket到object的所有信息
6. default.rgw.buckets.data：存放对象的数据，默认是副本池
7. default.rgw.buckets.non-ec：数据的额外信息存储池

8. default.rgw.users.uid：存放用户信息的存储池
9. default.rgw.data.root：存放bucket的元数据，结构体对应RGWBucketInfo，比如存放桶名、桶ID、data_pool等

重点对象存储（default.rgw）：

- 对象存储（Object Storage）是无层次结构的数据存储方法，通常用于云计算环境中
- **当创建RGW时候默认会创建对应的池信息**，不需要人为创建

1.2 查看pool的pg

docker exec rgw ceph osd pool get .rgw.root pg_num
pg_num: 32

docker exec rgw ceph osd pool get default.rgw.buckets.data pg_num

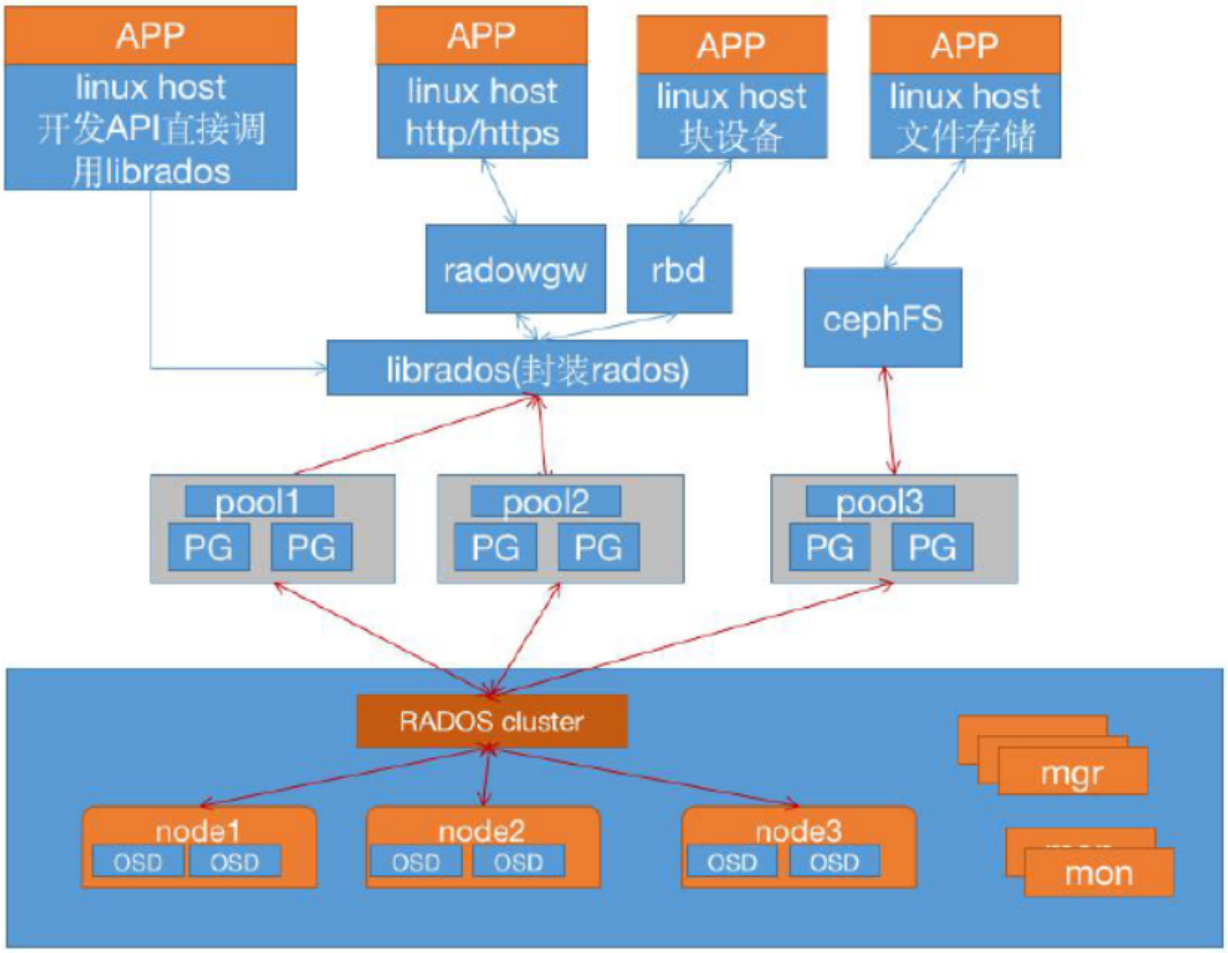
pg_num: 32

docker exec rgw ceph osd pool get cephfs_data pg_num
pg_num: 64

docker exec rgw ceph osd pool get cephfs_data pgp_num
pgp_num: 64

2 Ceph的核心组件及逻辑架构

2.1 逻辑框架



pool：pool资源池是Ceph存储数据的逻辑分区，**起到Namespace命名空间的作用**，不同的客户端可以去使用不同的pool资源池存储数据，pool除了可以隔离数据之外，还可以针对不同的pool资源池设置不同的优化策略，比如副本数、数据清洗次数、数据块及对象大小等等。

在pool资源池中会包含一定数量的PG，PG里的对象会被存储在不同的OSD中，pool资源上也是分布到整个集群。

PG(placement group)：

PG是一个逻辑概念，我们linux系统中可以直接看到对象，但是无法直接看到PG。它在数据寻址时类似于数据库中的索引：每个对象都会固定映射进一个PG中，所以当我们要寻找一个对象时，只需要先找到对象所属的PG，然后遍历这个PG就可以了，无需遍历所有对象。而且在数据迁移时，也是以PG作为基本单位进行迁移，ceph不会直接操作对象

一个pool 内部可以有多个PG 存在，pool 和PG 都是抽象的逻辑概念，一个pool 中有多少个PG 可以通过公式计算。引入PG这一层其实是为了更好的分配数据和定位数据。

PGP Placement Group for Placement purpose

- PG是指定存储池存储对象的目录有多少个，PGP是存储池PG的OSD分布组合个数。

pool资源池与PG载体的关系：

在前面说到过Object对象文件都是存储在OSD中的**PG目录中**，主要是为了故障迁移时，直接迁移目录会非常方便。

2.2 Ceph数据写入流程

在说明ceph数据写入流程之前之前，先介绍几个概念和它们之间的关系。

存储数据与object的关系：当用户要将数据存储到Ceph集群时，存储数据都会被分割成多个object，每个object都有一个object id，每个object的大小是可以设置的，默认是4MB，object可以看成是Ceph存储的最小存储单元。

object与pg的关系：由于object的数量很多，所以Ceph引入了pg的概念用于管理object，每个object最后都会通过CRUSH计算映射到某个pg中，**一个pg可以包含多个object。**

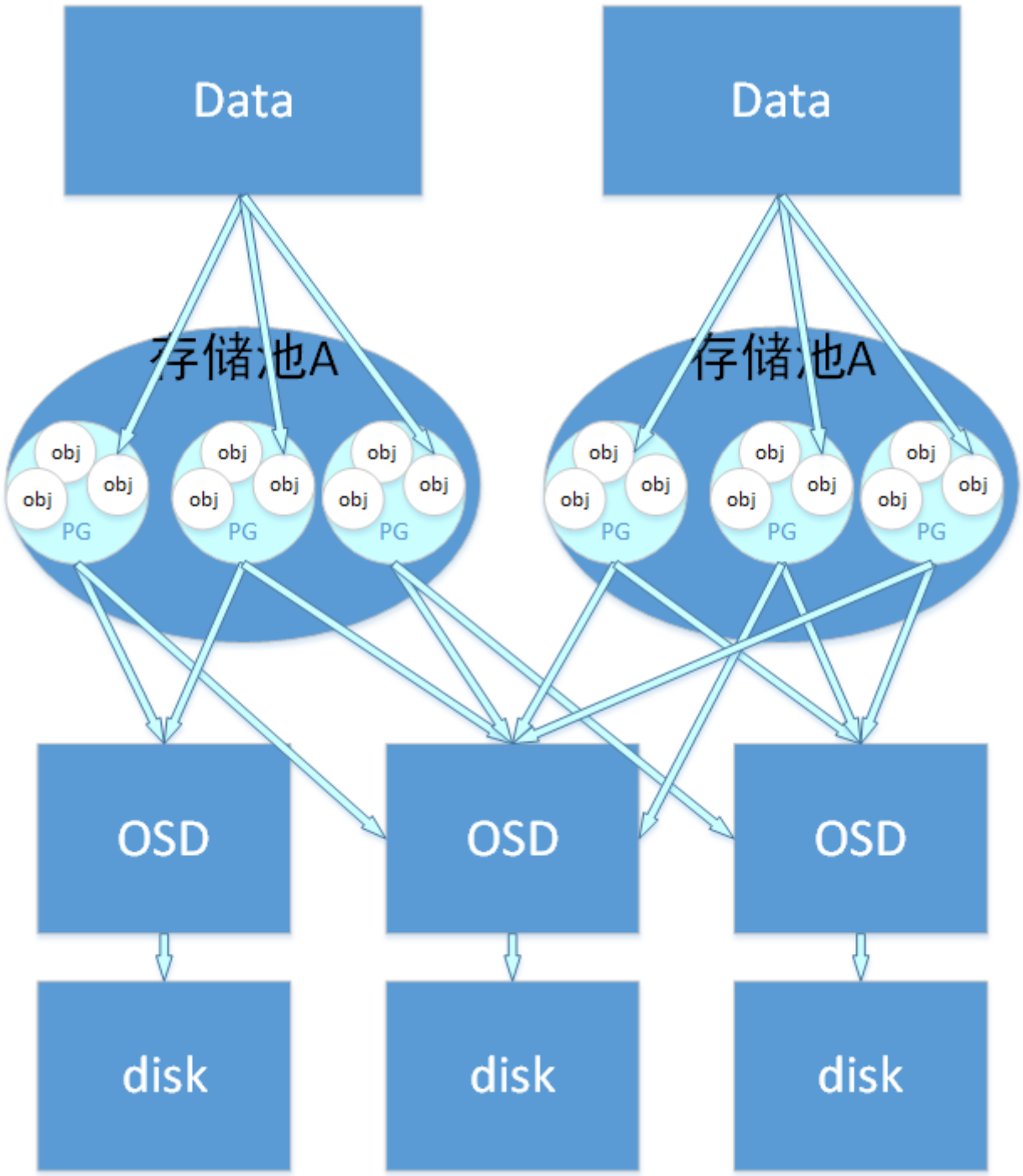
pg与osd的关系：pg也需要通过CRUSH计算映射到osd中去存储，**如果是二副本的，则每个pg都会映射到二个osd**，比如[osd.1,osd.2]，那么osd.1是存放该pg的**主副本**，osd.2是存放该**pg的从副本**，保证了数据的冗余。

pg和pgp的关系：pg是用来存放object的，**pgp**相当于是pg存放**osd的一种排列组合**，我举个例子，比如有3个osd，osd.1、osd.2和osd.3，副本数是2，如果pgp的数目为1，那么**pg存放的osd组合就只有一种**，可能是[osd.1,osd.2]，那么所有的pg主从副本分别存放到osd.1和osd.2，**如果pgp设为2**，那么其**osd组合可以两种**，可能是[osd.1,osd.2]和[osd.1,osd.3]，**一般来说应该将pg和pgp的数量设置为相等。**

pg和pool的关系：pool也是一个逻辑存储概念，我们创建存储池pool的时候，都需要指定pg和pgp的数量，逻辑上来说pg是属于某个存储池的，就有点像object是属于某个pg的

1. PG是指定存储池**存储对象**的目录有多少个，PGP是存储池PG的OSD分布组合个数
2. PG 和 Object 是一对多的关系，1个PG里面组织若干个Object，但是1个Object 只能被映射到1个PG中。
3. PG 和 OSD 是多对多的关系，1个PG会映射到多个OSD 上（按照副本或者纠删码规则），每个OSD也会承载多个PG
4. PG 和 pool 是多对一的关系。1个pool内包含多个PG，pool创建时可指定其中PG的数量。

以下这个图表明了存储数据，object、pg、pool、osd、存储磁盘的关系



ceph 集群部署好之后,要先创建存储池才能向ceph 写入数据，文件在向ceph 保存之前要先进行一致性hash计算，计算后会把文件保存在某个对应的PG 的，此文件一定属于某个pool 的一个PG，在通过PG 保存在OSD 上。数据对象在写到主OSD 之后再同步对从OSD 以实现数据的高可用。

2.3 pool资源池的基本操作

2.3.1 创建一个pool资源池

命令格式：`ceph osd create pool ${pool-name} ${pg_num} ${pgp_num}`

```
docker exec mon ceph osd pool create ceph-pool-1 64 64
```

在创建pool资源池的时候一定要指定pg_num和pgp_num参数，因为Ceph集群不能自动计算PG的数量。

官方建议的PG使用数量：

- 集群中小于5个OSD，则设置PG的数量为128。
- 集群有5-10个OSD时，设置PG的数量为512。
- 集群中有10-50个OSD时，设置PG的数量为1024.

当集群中超过50个OSD时，需要权衡PG的数量，有一种公式：

> PG 总数= (OSD 总数 x 100) /最大副本数
>
> 例如集群有90个OSD，pool设置的副本数为3：(90x100)/3=3000，然后结果必须舍入到最接近 2 的 N 次幂的值，所以设置为: **4096**

PGP的数量要和PG的数量保持一致。

注意这个PG数量是集群所能承担的总PG数量，每个资源池分配多个PG，还需要通过总的PG数乘以资源池占整个集群数据比例，最终拿到一个资源池应该分配多少个PG的数量。

2.3.2 为资源池设置应用模式

其实就是为资源池设置一个分类，有rbd、rgw、cephfs三种。

命令格式：`ceph osd pool application enable {pool_name} rbd`

```
docker exec mon ceph osd pool application enable ceph-pool-1 rbd
```

- RBD 全称 RADOS block device，是 Ceph 对外提供的块设备服务。
- RGW 全称 RADOS gateway，**是 Ceph 对外提供的对象存储服务**，**接口与 S3** 和 Swift 兼容。
- CephFS 全称 Ceph File System，是 Ceph 对外提供的文件系统服务。

2.3.3 查看资源池设置的应用模式

命令格式：ceph osd pool application get {pool_name}

```
docker exec mon ceph osd pool application get ceph-pool-1
```

2.3.4 查看OSD所有的资源池

```
docker exec mon ceph osd lspools
```

2.3.5 查看资源池的PG数量和PGP数量

查看pool资源池属性的命令格式；`ceph osd pool get {poll_name} [parameter]`

```
 查看pg的数量
docker exec mon ceph osd pool get ceph-pool-1 pg_num

 查看pgp的数量
docker exec mon ceph osd pool get ceph-pool-1 pgp_num
```

2.3.6 查看资源池的副本数量

副本数量默认为3个。

```
docker exec mon ceph osd pool get ceph-pool-1 size

docker exec mon ceph osd pool get default.rgw.buckets.data size
```

2.3.7 查看资源池的类型

默认类型为replicated_rule（复制类型）。

```
docker exec mon ceph osd pool get ceph-pool-1 crush_rule

docker exec mon ceph osd pool get default.rgw.buckets.data crush_rule
```

2.3.8 设置资源池的PG数量以及PGP数量

修改pool资源池属性的命令格式；`ceph osd pool set {poll_name} [parameter]`

```
docker exec mon ceph osd pool set ceph-pool-1 pg_num 128

docker exec mon ceph osd pool set ceph-pool-1 pgp_num 128
```

2.3.9 设置资源池的副本数量

```
docker exec mon ceph osd pool set ceph-pool-1 size 2
```

2.3.10 设置资源池的最大object对象数量

当我们有很多存储池的时候，有些作为公共存储池，这时候就有必要为这些存储池做一些配额，**限制可存放的文件数**，或者空间大小，以免无限的增大影响到集群的正常运行

命令格式：`ceph osd pool set-quota {pool_name} [max_objects {obj-count}] [max_bytes {bytes}]`

```
docker exec mon ceph osd pool set-quota ceph-pool-1 max_objects 10000
```

不能超过磁盘。

2.3.11 重命名资源池

命令格式：`ceph osd pool rename {current-pool-name} {new-pool-name}`

```
docker exec mon ceph osd pool rename ceph-pool-1 ceph-pool-2
```

2.3.12 查看资源池的统计信息

```
docker exec mon ceph df
```

```
lqf@master:~$ docker exec mon ceph df
RAW STORAGE:
    CLASS     SIZE        AVAIL       USED        RAW USED     %RAW USED
    hdd       300 GiB     296 GiB     527 MiB      3.5 GiB          1.17
    TOTAL     300 GiB     296 GiB     527 MiB      3.5 GiB          1.17

POOLS:
    POOL                            ID     PGS     STORED      OBJECTS     USED
%USED     MAX AVAIL
    .rgw.root                        1      32     1.2 KiB           4     768 KiB
      0        94 GiB
    default.rgw.control              2      32         0 B           8         0 B
      0        94 GiB
    default.rgw.meta                 3      32     1.2 KiB           7     1.1 MiB
      0        94 GiB
```

```
    default.rgw.log                 4      32         0 B         207         0 B
      0        94 GiB
    cephfs_data                     5      64         0 B           0         0 B
      0        94 GiB
    cephfs_metadata                 6      32     3.2 KiB          22     1.5 MiB
      0        94 GiB
    default.rgw.buckets.index       7      32         0 B           2         0 B
      0        94 GiB
    default.rgw.buckets.data        8      32     102 MiB          29     307 MiB
    0.11        94 GiB
    default.rgw.buckets.non-ec      9      32         0 B           0         0 B
      0        94 GiB
    ceph-pool-2                    10     128         0 B           0         0 B
      0       141 GiB
```

2.3.13 查看资源池的利用率

```
docker exec mon rados df
```

2.3.14 删除资源池

```
docker exec mon ceph osd pool delete ceph-pool-2 --yes-i-really-really-mean-it

报错：Error EPERM: WARNING: this will *PERMANENTLY DESTROY* all data stored in pool
testpool.  If you are *ABSOLUTELY CERTAIN* that is what you want, pass the pool name
*twice*, followed by --yes-i-really-really-mean-it.
```

提示：删除存储池命令存在数据丢失的风险，**Ceph于是默认禁止此类操作。管理员需要在ceph.conf配置文件中启用支持删除存储池**的操作后，方可使用类似上述命令删除存储池；

2.3.15 制作池的快照

要创建池的快照，请执行：ceph osd pool mksnap {pool-name} {snap-name}

```
docker exec mon ceph osd pool mksnap {pool-name} {snap-name}
```

快照 消耗空间

2.3.16 删除池的快照

要删除池的快照，请执行：ceph osd pool rmsnap {pool-name} {snap-name}

```
docker exec mon ceph osd pool rmsnap {pool-name} {snap-name}
```

2.4 测试上传下载数据对象

2.4.1 创建存储池并设置PG数量为16个

docker exec mon ceph osd pool create testpool 16 16

查看：

docker exec mon ceph osd pool ls

2.4.2 上传文件到testpool

docker exec mon rados put test /etc/profile -p testpool

docker exec mon rados ls -p testpool

提示：可以看到我们上传mon容器的/etc/profile 文件到testpool存储池中并命名为test，对应文件已将在testpool存储中存在；说明上传没有问题；

2.4.3 获取存储池中数据对象的具体位置信息

```
docker exec mon ceph osd map testpool test

显示：
osdmap e1294 pool 'testpool' (11) object 'test' -> pg 11.40e8aab5 (11.5) -> up
([1,0,2], p1) acting ([1,0,2], p1)
```

提示：可以看到test文件在testpool存储中被分别存放编号为1、0、2的osd上去了；

2.4.4 下载文件到本地

```
docker exec mon rados get test test-down -p testpool

实际上是下载到mon镜像里了。
```

2.4.5 删除数据对象

```
docker exec mon rados rm test -p testpool
docker exec mon rados ls -p testpool
```

2.4.6 删除存储池

```
docker exec mon ceph osd pool rm testpool --yes-i-really-really-mean-it.
显示:
Error EPERM: WARNING: this will *PERMANENTLY DESTROY* all data stored in pool
testpool.  If you are *ABSOLUTELY CERTAIN* that is what you want, pass the pool name
*twice*, followed by --yes-i-really-really-mean-it.
```

提示：删除存储池命令存在数据丢失的风险，Ceph于是默认禁止此类操作。管理员需要在ceph.conf配置文件中启用支持删除存储池的操作后，方可使用类似上述命令删除存储池；

3 ceph经验

当Ceph集群存储容量快接近水位， 扩容一次扩多少节点合适?

- 设置好水位线一般70%以内，然后进行扩容， 扩容的时候会有数据发送迁移， 控制好集群内数据迁移的大小， 避免迁移量太大， 影响客户端io请求延迟。
- 单个节点方式逐步增加，待数据平衡完成后在进行下一个节点，这样影响较小，比较安全。
- 当您计划购买或升级当前 `Ceph` 中的 `OSD` 前，最重要的是要根据当前状况进行数据量预测，以匹配未来生产的数据量。通常，最好至**少提前6到12个月进行估算**，并将此存储量乘以所需的对象冗余量（即 `32TB` 数据* `3` （副本数）= `96TB` 所需的存储空间）

调整pg数量有什么问题需要注意，会有什么影响，怎样提前规划好?

前期由于存储容量和集群规模不大，设置的pg数量也比较小，后期由于集群规模变大，势必需要调整pg数量，调整pg数量有什么问题需要注意，会有什么影响，怎样提前规划好?

- 调整pg数目时会发生大量pg分裂迁移，建议在业务低峰期进行并做好恢复带宽限制。如果集群不能一次规划建设到位的话建议按照官方算法按照每次扩容完成后的总osd数目进行调整以获得最佳配比。
- 计算公式

  （1）预设Ceph集群中的PG数至关重要，公式如下:

PG 总数 = (OSD 数 100) / 最大副本数

（2） 集群中单个池的PG数计算公式如下

PG 总数 = (OSD 数 100) / 最大副本数 / 池数

Ceph扩容后，集群内数据迁移量过大，怎样不影响用户IO阻塞?

- 可以限制重平衡的io数目和重平衡的io带宽减少对正常业务io的影响，或者只在业务低峰期打开重平衡。
- 降低recovery的I/O优先级，使得Client IO优先，即使带宽优先供给客户端。
- 通过对数据迁移进行时间和流量上的双重QOS策略来保证线上业务稳定运行。
- 集群内部流量迁移进行速率的控制， 优先保证客户端io请求流量。比如限制内网路由的带宽。

ceph集群扩容前要做哪些准备工作，如何确保扩容过程对业务影响最小?

- 建议在规划存储池时对存储池的规模容量做好规划，一次建设完成，以避免对现有存储池的扩容，现有存储池扩容时会发生大量的数据迁移，迁移整体耗时会较长，如果必须扩容的话，可以提前评估业务低峰窗口，并限制数据平衡速率，在扩容完成后根据osd数目调整pg数目，减少对业务io的影响。
- 扩容前，需要评估内部集群数据的迁移量，做好集群内部流量迁移的限速，避免迁移量太大，影响客户端io请求。

osd 启动报错了，应该从哪几个方面诊断排错?

- 一般报错的提示都是比较明晰的，按照提示处理就可以了。
- 然后根据问题，找对应的解决方案，一般网上都有对应的解决方案。最后， 搞不定把**详细的步骤和日志**，帖到社区。
- 原因太多了，主要工具就是看日志：

（1）文件系统的问题；

（2）认证的问题；

（3）osd状态不对；

（4）osd资源问题

ceph如何进行性能优化?

性能优化从底层硬件、网络、操作系统、软件参数、缓存等几方面

一、硬件：选用ceph合适的硬件。尽量增加SSD，合理分配到pool和index

二、网络：1.增加收发包ethtool -G eth4 rx 8192 tx 8192

2.增加网络带宽，区分集群内外网络

三、操作系统：1.调整包 echo "8192”> /sys/block/sda/queue/read_ahead_kb

2.减少swap使用vm.swappiness = 5

四、软件:

1.RGW（对象存储）

rgw_cache_enabled = true # 开启RGW cache

rgw_thread_pool_size = 2000

rgw_cache_lru_size = 20000

rgw_num_rados_handles = 128

2.RBD（块存储）

rbd_cache_enabled = true # 开启RBD cache

rbd_cache_size = 268435456

rbd_cache_max_dirty = 134217728

rbd_cache_max_dirty_age = 5

3.优化OSD scrub周期及相关参数

osd scrub begin hour = 4

osd scrub end hour = 12

osd scrub chunk min = 5

osd scrub chunk max = 5

osd scrub sleep = 1（不建议增大太多，会延长scrub完成时间）

osd scrub min interval = 2592000

osd scrub max interval = 2592000

osd deep scrub interval = 2419200

osd scrub load threshold = 0.30

osd scrub during recovery = false

osd scrub priority = 5

osd scrub interval randomize ratio = 0.35

五、缓存：增加磁盘缓存到SSD上（有版本要求） ceph-volume lvmcache add

RADOS网关冗余不足

如果通过 RADOS 网关访问您的 Ceph 集群，那么促进 API 访问的前端服务器必须冗余且能够负载均衡，这一点非常重要。

配置多个网关服务。

硬件配置不足

为 CEPH 集群维护硬件时，最重要的是要确保硬件配置满足实际需求，具体需要考虑的如下：

- **「电源」**：确保主机的电源模块至少是双路冗余。还要确保每台 CEPH 服务器都有一个备用电源，该备用电源的功率足以完全满足服务器的需求。没有适当的冗余，您将面临不可挽回的数据丢失的风险。
- **「CPU」**：最佳实践要求在所有 CEPH 服务器上使用相同型号相同配置的 CPU。这有助于在整个 ceph 集群中保持一致性以及稳定性。
- **「内存」**：与 CPU 相似，您使用的内存应在 CEPH 服务器之间平均分配。理想情况下，存储服务器的品牌和规格应该相同。此外，在发生硬件故障时，还应具有大量可用的冗余内存（备件）。
- **「硬盘」**：建议将 SAS 磁盘用于 OSD。如果有可能的话，使用故障率低于 SAS 盘的 NVMe 磁盘则会更为理想。
- **「其他」**：如果有可能的话，可以提供一些冷备机器。当存储节点存现故障的时候，极端情况下可以直接进行备机更换（保留物理磁盘）。

硬盘类型	数据传输速度	IOPS	延迟
SAS	高	高	较低
SATA	中等	中等	中等
NVMe	非常高	非常高	非常低

4 参考

1. Ceph之PG数调整 - TuringM - 博客园 (cnblogs.com)
2. Ceph集群CephFS文件存储核心概念及部署使用详解*其它综合脚本之家* (jb51.net)
3. Ceph
4. Ceph对象存储使用-CSDN博客
5. 一、Ceph基础及工作原理 - zhrx - 博客园 (cnblogs.com)
6. 基于生产经验总结出的Ceph使用案例 (qq.com)
7. Ceph 日常运维常见难点和故障的解决办法 (qq.com)
8. ceph 中的 CRUSH 算法 - 简书 (jianshu.com)