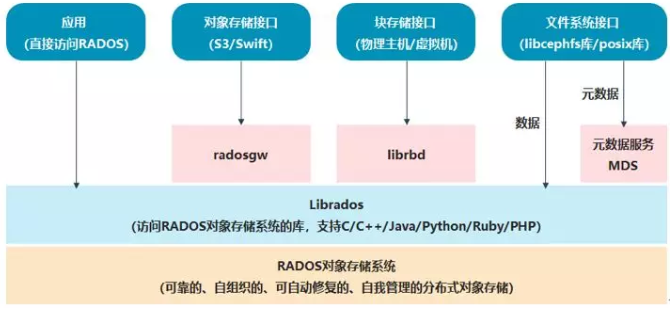
**[Ceph分布式存储工作原理 及 部署介绍](https://www.cnblogs.com/kevingrace/p/8387999.html)**

存储根据其类型，可分为块存储，对象存储和文件存储。在主流的分布式存储技术中，HDFS/GPFS/GFS属于文件存储，Swift属于对象存储，而Ceph可支持块存储、对象存储和文件存储，故称为统一存储。

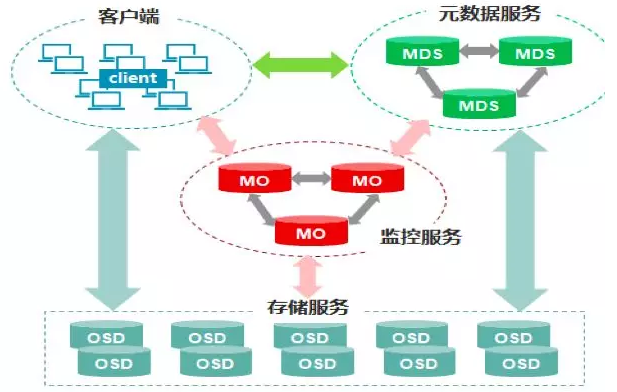
**一、Ceph 基本介绍**  
Ceph是一个分布式存储系统，诞生于2004年，最早致力于开发下一代高性能分布式文件系统的项目。经过多年的发展之后，已得到众多云计算和存储厂商的支持，成为应用最广泛的开源分布式存储平台。Ceph源码下载：[http://ceph.com/download/](http://ceph.com/download/%20" \t "_blank)。随着云计算的发展，ceph乘上了OpenStack的春风，进而成为了开源社区受关注较高的项目之一。Ceph可以将多台服务器组成一个超大集群，把这些机器中的磁盘资源整合到一块儿，形成一个大的资源池（PB级别），然后按需分配给应用使用。

**1.  Ceph的主要架构**



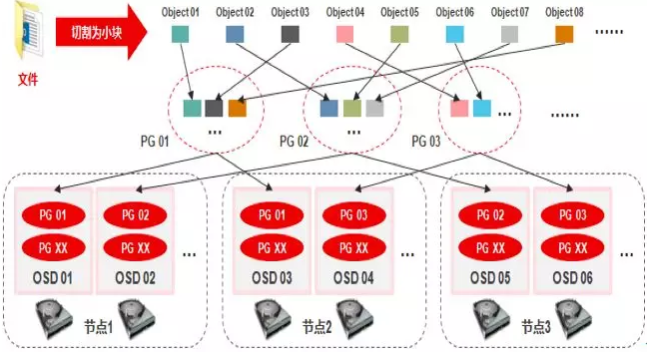
**<1>**Ceph的最底层是RADOS（分布式对象存储系统），它具有可靠、智能、分布式等特性，实现高可靠、高可拓展、高性能、高自动化等功能，并最终存储用户数据。RADOS系统主要由两部分组成，分别是OSD和Monitor。  
**<2>**RADOS之上是LIBRADOS，LIBRADOS是一个库，它允许应用程序通过访问该库来与RADOS系统进行交互，支持多种编程语言，比如C、C++、Python等。  
**<3>**基于LIBRADOS层开发的有三种接口，分别是RADOSGW、librbd和MDS。  
**<4>**RADOSGW是一套基于当前流行的RESTFUL协议的网关，支持对象存储，兼容S3和Swift。  
**<5>**librbd提供分布式的块存储设备接口，支持块存储。  
**<6>**MDS提供兼容POSIX的文件系统，支持文件存储。

**2.  Ceph的功能模块**

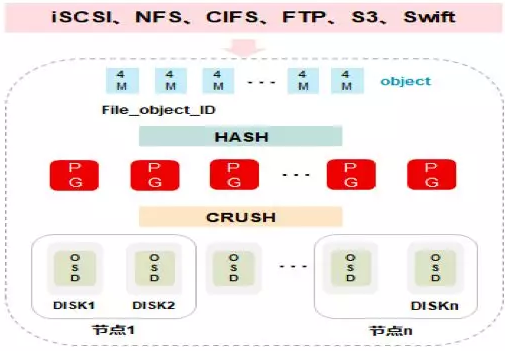


Ceph的核心组件包括Client客户端、MON监控服务、MDS元数据服务、OSD存储服务，各组件功能如下：  
**<1>**Client客户端：负责存储协议的接入，节点负载均衡。  
**<2>**MON监控服务：负责监控整个集群，维护集群的健康状态，维护展示集群状态的各种图表，如OSD Map、Monitor Map、PG Map和CRUSH Map。  
**<3>**MDS元数据服务：负责保存文件系统的元数据，管理目录结构。  
**<4>**OSD存储服务：主要功能是存储数据、复制数据、平衡数据、恢复数据，以及与其它OSD间进行心跳检查等。一般情况下一块硬盘对应一个OSD。

**3.  Ceph的资源划分**  
Ceph采用crush算法，在大规模集群下，实现数据的快速、准确存放，同时能够在硬件故障或扩展硬件设备时，做到尽可能小的数据迁移，其**原理如下**：  
**<1>**当用户要将数据存储到Ceph集群时，数据先被分割成多个object，(每个object一个object id，大小可设置，默认是4MB），object是Ceph存储的最小存储单元。  
**<2>**由于object的数量很多，为了有效减少了Object到OSD的索引表、降低元数据的复杂度，使得写入和读取更加灵活，引入了pg(Placement Group )：PG用来管理object，每个object通过Hash，映射到某个pg中，一个pg可以包含多个object。  
**<3>**Pg再通过CRUSH计算，映射到osd中。如果是三副本的，则每个pg都会映射到三个osd，保证了数据的冗余。



**4.  Ceph的数据写入**  
Ceph数据的写入流程  
**<1>**数据通过负载均衡获得节点动态IP地址；  
**<2>**通过块、文件、对象协议将文件传输到节点上；  
**<3>**数据被分割成4M对象并取得对象ID；  
**<4>**对象ID通过HASH算法被分配到不同的PG；  
**<5>**不同的PG通过CRUSH算法被分配到不同的OSD



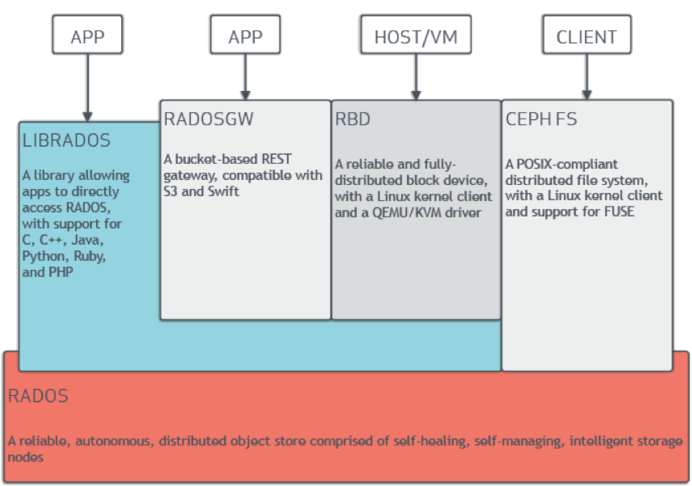
**5.  Ceph的特点**  
**<1>**Ceph支持对象存储、块存储和文件存储服务，故称为统一存储。  
**<2>**采用CRUSH算法，数据分布均衡，并行度高，不需要维护固定的元数据结构；  
**<3>**数据具有强一致，确保所有副本写入完成才返回确认，适合读多写少场景；  
**<4>**去中心化，MDS之间地位相同，无固定的中心节点

Ceph存在一些缺点  
**<1>**去中心化的分布式解决方案，需要提前做好规划设计，对技术团队的要求能力比较高。  
**<2>**Ceph扩容时，由于其数据分布均衡的特性，会导致整个存储系统性能的下降。

Ceph相比于其他存储方案的优势  
**<1>**CRUSH算法：Crush算法是ceph的两大创新之一，简单来说，Ceph摒弃了传统的集中式存储元数据寻址的方案，转而使用CRUSH算法完成数据的寻址操作。CRUSH在一致性哈希基础上很好的考虑了容灾域的隔离，能够实现各类负载的副本放置规则，例如跨机房、机架感知等。Crush算法有相当强大的扩展性，理论上支持数千个存储节点。  
**<2>**高可用：Ceph中的数据副本数量可以由管理员自行定义，并可以通过CRUSH算法指定副本的物理存储位置以分隔故障域，支持数据强一致性； Ceph可以忍受多种故障场景并自动尝试并行修复；Ceph支持多份强一致性副本，副本能够垮主机、机架、机房、数据中心存放。所以安全可靠。Ceph存储节点可以自管理、自动修复。无单点故障，容错性强。  
**<3>**高性能：因为是多个副本，因此在读写操作时候能够做到高度并行化。理论上，节点越多，整个集群的IOPS和吞吐量越高。另外一点Ceph客户端读写数据直接与存储设备(osd) 交互。在块存储和对象存储中无需元数据服务器。  
**<4>**高扩展性：Ceph不同于Swift，客户端所有的读写操作都要经过代理节点。一旦集群并发量增大时，代理节点很容易成为单点瓶颈。Ceph本身并没有主控节点，扩展起来比较容易，并且理论上，它的性能会随着磁盘数量的增加而线性增长。Ceph扩容方便、容量大。能够管理上千台服务器、EB级的容量。  
**<5>**特性丰富：Ceph支持三种调用接口：对象存储，块存储，文件系统挂载。三种方式可以一同使用。在国内一些公司的云环境中，通常会采用Ceph作为openstack的唯一后端存储来提升数据转发效率。Ceph是统一存储，虽然它底层是一个分布式文件系统，但由于在上层开发了支持对象和块的接口，所以在开源存储软件中，优势很明显。

Ceph提供3种存储方式分别是对象存储，块存储和文件系统，一般我们主要关心的还是块存储，推荐将虚拟机后端存储从SAN过渡到Ceph。Ceph 现在是云计算、虚拟机部署的最火开源存储解决方案，据统计大概有20%的OpenStack部署存储用的都是Ceph的block storage。

**二、Ceph架构详解**  
Ceph根据场景可分为对象存储、块设备存储和文件存储。Ceph相比其它分布式存储技术，其优势点在于：它不单是存储，同时还充分利用了存储节点上的计算能力，在存储每一个数据时，都会通过计算得出该数据存储的位置，尽量将数据分布均衡。同时，由于采用了CRUSH、HASH等算法，使得它不存在传统的单点故障，且随着规模的扩大，性能并不会受到影响。



**<1>**Ceph的底层是RADOS，RADOS本身也是分布式存储系统，CEPH所有的存储功能都是基于RADOS实现。RADOS采用C++开发，所提供的原生Librados API包括C和C++两种。Ceph的上层应用调用本机上的librados API，再由后者通过socket与RADOS集群中的其他节点通信并完成各种操作。  
**<2>**RADOS向外界暴露了调用接口，即LibRADOS,应用程序只需要调用LibRADOS的接口，就可以操纵Ceph了。这其中，RADOS GW用于对象存储，RBD用于块存储，它们都属于LibRADOS;CephFS是内核态程序，向外界提供了POSIX接口，用户可以通过客户端直接挂载使用。  
**<3>**RADOS GateWay、RBD其作用是在librados库的基础上提供抽象层次更高、更便于应用或客户端使用的上层接口。其中，RADOS GW是一个提供与Amazon S3和Swift兼容的RESTful API的gateway，以供相应的对象存储应用开发使用。RBD则提供了一个标准的块设备接口，常用于在虚拟化的场景下为虚拟机创建volume。目前，Red Hat已经将RBD驱动集成在KVM/QEMU中，以提高虚拟机访问性能。这两种方式目前在云计算中应用的比较多。  
**<4>**CEPHFS则提供了POSIX接口，用户可直接通过客户端挂载使用。它是内核态程序，所以无需调用用户空间的librados库。它通过内核中net模块来与Rados进行交互。

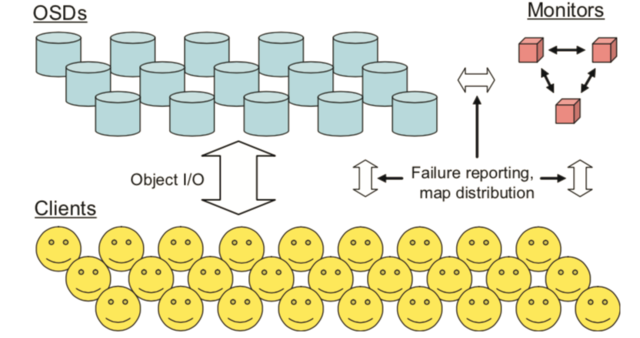
**1.  Ceph之RADOS说明**  
RADOS (Reliable, Autonomic Distributed Object Store) 是Ceph的核心之一，作为Ceph分布式文件系统的一个子项目，特别为Ceph的需求设计，能够在动态变化和异质结构的存储设备机群之上提供一种稳定、可扩展、高性能的单一逻辑对象(Object)存储接口和能够实现节点的自适应和自管理的存储系统。在传统分布式存储架构中，存储节点往往仅作为被动查询对象来使用，随着存储规模的增加，数据一致性的管理会出现很多问题。而新型的存储架构倾向于将基本的块分配决策和安全保证等操作交给存储节点来做，然后通过提倡客户端和存储节点直接交互来简化数据布局并减小io瓶颈。

RADOS就是这样一个可用于PB级规模数据存储集群的可伸缩的、可靠的对象存储服务。它包含两类节点：存储节点、管理节点。它通过利用存储设备的智能性，将诸如一致性数据访问、冗余存储、错误检测、错误恢复分布到包含了上千存储节点的集群中，而不是仅仅依靠少数管理节点来处理。

RADOS中的存储节点被称为OSD(object storage device)，它可以仅由很普通的组件来构成，只需要包含CPU、网卡、本地缓存和一个磁盘或者RAID，并将传统的块存储方式替换成面向对象的存储。在PB级的存储规模下，存储系统一定是动态的：系统会随着新设备的部署和旧设备的淘汰而增长或收缩，系统内的设备会持续地崩溃和恢复，大量的数据被创建或者删除。

RADOS通过 cluster map来实现这些，cluster map会被复制到集群中的所有部分（存储节点、控制节点，甚至是客户端），并且通过怠惰地传播小增量更新而更新。Cluster map中存储了整个集群的数据的分布以及成员。通过在每个存储节点存储完整的Cluster map，存储设备可以表现的半自动化，通过peer-to-peer的方式（比如定义协议）来进行数据备份、更新，错误检测、数据迁移等等操作。这无疑减轻了占少数的monitor cluster（管理节点组成的集群）的负担。

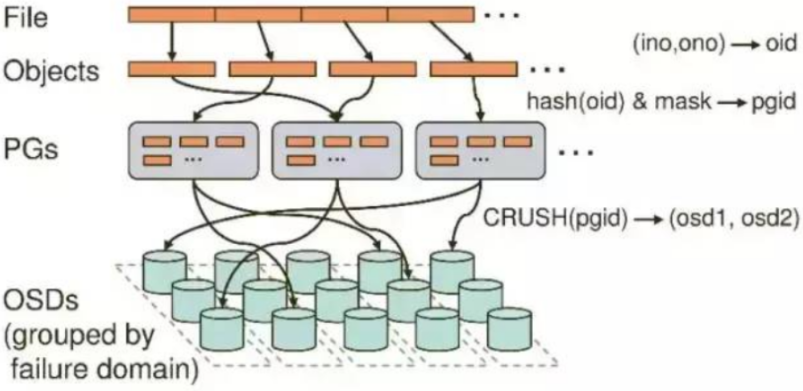
**RADOS设计如下：**



一个RADOS系统包含大量的OSDs 和 很少的用于管理OSD集群成员的monitors。OSD的组成如简介所说。而monitor是一些独立的进程，以及少量的本地存储，monitor之间通过一致性算法保证数据的一致性。

**Cluster Map**  
存储节点集群通过monitor集群操作cluster map来实现成员的管理。cluster map 描述了哪些OSD被包含进存储集群以及所有数据在存储集群中的分布。cluster map不仅存储在monitor节点，它被复制到集群中的每一个存储节点，以及和集群交互的client。当因为一些原因，比如设备崩溃、数据迁移等，cluster map的内容需要改变时，cluster map的版本号被增加，map的版本号可以使通信的双方确认自己的map是否是最新的，版本旧的一方会先将map更新成对方的map，然后才会进行后续操作。

**Data Placement**  
下面总体说下RADOS的存储层次，RADOS中基本的存储单位是对象，一般为2MB或4MB，当一个文件要存入RADOS时，首先会被切分成大小固定的对象（最后一个对象大小可能不同），然后将对象分配到一个PG（Placement Group）中，然后PG会复制几份，伪随机地派给不同的存储节点。当新的存储节点被加入集群，会在已有数据中随机抽取一部分数据迁移到新节点。这种概率平衡的分布方式可以保证设备在潜在的高负载下正常工作。更重要的是，数据的分布过程仅需要做几次随机映射，不需要大型的集中式分配表。如下图是Ceph内部架构：



对于每个层次的详细说明：  
**<1>**File—— 用户需要存储或者访问的文件。  
**<2>**Object—— RADOS的基本存储单元。Object与上面提到的file的区别是，object的最大size由RADOS限定（通常为2MB或4MB），以便实现底层存储的组织管理。因此，当上层应用向RADOS存入size很大的file时，需要将file切分成统一大小的一系列object（最后一个的大小可以不同）进行存储。  
**<3>**PG（Placement Group）—— 对object的存储进行组织和位置映射。具体而言，一个PG负责组织若干个object（可以为数千个甚至更多），但一个object只能被映射到一个PG中，即，PG和object之间是“一对多”映射关系。同时，一个PG会被映射到n个OSD上，而每个OSD上都会承载大量的PG，即，PG和OSD之间是“多对多”映射关系。在实践当中，n至少为2（n代表冗余的份数），如果用于生产环境，则至少为3。一个OSD上的PG则可达到数百个。事实上，PG数量的设置牵扯到数据分布的均匀性问题。  
**<4>**OSD—— 即object storage device，前文已经详细介绍，此处不再展开。唯一需要说明的是，OSD的数量事实上也关系到系统的数据分布均匀性，因此其数量不应太少。在实践当中，至少也应该是数十上百个的量级才有助于Ceph系统的设计发挥其应有的优势。

各层次之间的映射关系：  
**<1>**file -> object  
object的最大size是由RADOS配置的，当用户要存储一个file，需要将file切分成几个object。  
**<2>**object -> PG  
每个object都会被映射到一个PG中，然后以PG为单位进行备份以及进一步映射到具体的OSD上。  
**<3>**PG -> OSD  
根据用户设置的冗余存储的个数r，PG会最终存储到r个OSD上，这个映射是通过一种伪随机的映射算法 CRUSH 来实现的，这个算法的特点是可以进行配置。

**Ceph存储过程描述**  
每台服务器都有好几块磁盘（sda,sdb,sdc等），磁盘又可以进一步分区（sda1,sda2等）。Ceph中最基本的进程就是OSD(对象存储设备)，每个磁盘对应一个OSD。如果用户通过客户端想要存储一个文件，那么在RADOS中，该文件实际上会分为一个个4M块大小的对象。每个文件都一个文件ID(例如A),那么这些对象的ID就是（A0,A1,A2等）。然而在分布式储存系统中，有成千上万个对象，光遍历就要花很长的时间，所以对象会先通过hash-取模运算，存放到一个PG（Place Group）中，PG相当于数据库中的索引（PG的数量是固定的，不会随着OSD的增加或者删除而改变），这样只需要首先定位到PG位置，然后在PG中查询对象即可，大大提高了查询效率。之后PG中的对象又会根据设置的副本数量进行复制，并根据Crush算法存储到OSD节点上。

无论使用哪种存储方式（对象、块、挂载），存储的数据都会被切分成对象（Objects）。Objects size大小可以由管理员调整，通常为2M或4M。每个对象都会有一个唯一的OID，由ino与ono生成，虽然这些名词看上去很复杂，其实相当简单。ino即是文件的File ID，用于在全局唯一标示每一个文件，而ono则是分片的编号。比如：一个文件FileID为A，它被切成了两个对象，一个对象编号0，另一个编号1，那么这两个文件的oid则为A0与A1。Oid的好处是可以唯一标示每个不同的对象，并且存储了对象与文件的从属关系。由于ceph的所有数据都虚拟成了整齐划一的对象，所以在读写时效率都会比较高。

但是对象并不会直接存储进OSD中，因为对象的size很小，在一个大规模的集群中可能有几百到几千万个对象。这么多对象光是遍历寻址，速度都是很缓慢的；并且如果将对象直接通过某种固定映射的哈希算法映射到osd上，当这个osd损坏时，对象无法自动迁移至其他osd上面（因为映射函数不允许）。为了解决这些问题，ceph引入了归置组的概念，即PG。

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。

对象时如何映射进PG的？还记得OID么？首先使用静态hash函数对OID做hash取出特征码，用特征码与PG的数量去模，得到的序号则是PGID。由于这种设计方式，PG的数量多寡直接决定了数据分布的均匀性，所以合理设置的PG数量可以很好的提升CEPH集群的性能并使数据均匀分布。

最后PG会根据管理员设置的副本数量进行复制，然后通过crush算法存储到不同的OSD节点上（其实是把PG中的所有对象存储到节点上），第一个osd节点即为主节点，其余均为从节点。

**Ceph集群维护**  
前面已经介绍了，由若干个monitor共同负责整个RADOS集群中所有OSD状态的发现与记录，并且共同形成cluster map的master版本，然后扩散至全体OSD以及client。OSD使用Cluster map进行数据的维护，而client使用Cluster map进行数据的寻址。monitor并不主动轮询各个OSD的当前状态。相反，OSD需要向monitor上报状态信息。常见的上报有两种情况：一是新的OSD被加入集群，二是某个OSD发现自身或者其他OSD发生异常。在收到这些上报信息后，monitor将更新cluster map信息并加以扩散。

Cluster map的实际内容包括：  
**<1>**Epoch，即版本号。cluster map的epoch是一个单调递增序列。epoch越大，则cluster map版本越新。因此，持有不同版本cluster map的OSD或client可以简单地通过比较epoch决定应该遵从谁手中的版本。而monitor手中必定有epoch最大、版本最新的cluster map。当任意两方在通信时发现彼此epoch值不同时，将默认先将cluster map同步至高版本一方的状态，再进行后续操作。  
**<2>**各个OSD的网络地址。  
**<3>**各个OSD的状态。OSD状态的描述分为两个维度：up或者down（表明OSD是否正常工作），in或者out（表明OSD是否在至少一个PG中）。因此，对于任意一个OSD，共有四种可能的状态：  
-   up且in：说明该OSD正常运行，且已经承载至少一个PG的数据。这是一个OSD的标准工作状态；  
-   up且out：说明该OSD正常运行，但并未承载任何PG，其中也没有数据。一个新的OSD刚刚被加入Ceph集群后，便会处于这一状态。而一个出现故障的OSD被修复后，重新加入Ceph集群时，也是处于这一状态；  
-   down且in：说明该OSD发生异常，但仍然承载着至少一个PG，其中仍然存储着数据。这种状态下的OSD刚刚被发现存在异常，可能仍能恢复正常，也可能会彻底无法工作；  
-   down且out：说明该OSD已经彻底发生故障，且已经不再承载任何PG。  
**<4>**CRUSH算法配置参数。表明了Ceph集群的物理层级关系（cluster hierarchy），位置映射规则（placement rules）。  
根据cluster map的定义可以看出，其版本变化通常只会由"3"和"4"两项信息的变化触发。而这两者相比，"3"发生变化的概率更高一些。

一个新的OSD上线后，首先根据配置信息与monitor通信。Monitor将其加入cluster map，并设置为up且out状态，再将最新版本的cluster map发给这个新OSD。收到monitor发来的cluster map之后，这个新OSD计算出自己所承载的PG（为简化讨论，此处我们假定这个新的OSD开始只承载一个PG），以及和自己承载同一个PG的其他OSD。然后，新OSD将与这些OSD取得联系。如果这个PG目前处于降级状态（即承载该PG的OSD个数少于正常值，如正常应该是3个，此时只有2个或1个。这种情况通常是OSD故障所致），则其他OSD将把这个PG内的所有对象和元数据复制给新OSD。数据复制完成后，新OSD被置为up且in状态。而cluster map内容也将据此更新。这事实上是一个自动化的failure recovery过程。当然，即便没有新的OSD加入，降级的PG也将计算出其他OSD实现failure recovery。

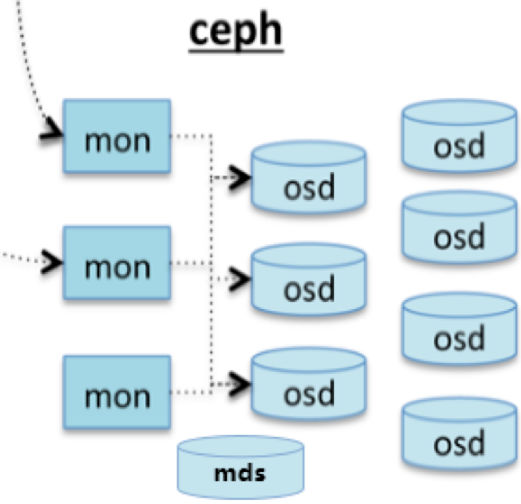
如果该PG目前一切正常，则这个新OSD将替换掉现有OSD中的一个（PG内将重新选出Primary OSD），并承担其数据。在数据复制完成后，新OSD被置为up且in状态，而被替换的OSD将退出该PG（但状态通常仍然为up且in，因为还要承载其他PG）。而cluster map内容也将据此更新。这事实上是一个自动化的数据re-balancing过程。如果一个OSD发现和自己共同承载一个PG的另一个OSD无法联通，则会将这一情况上报monitor。此外，如果一个OSD deamon发现自身工作状态异常，也将把异常情况主动上报给monitor。在上述情况下，monitor将把出现问题的OSD的状态设为down且in。如果超过某一预订时间期限，该OSD仍然无法恢复正常，则其状态将被设置为down且out。反之，如果该OSD能够恢复正常，则其状态会恢复为up且in。在上述这些状态变化发生之后，monitor都将更新cluster map并进行扩散。这事实上是自动化的failure detection过程。

对于一个RADOS集群而言，即便由数千个甚至更多OSD组成，cluster map的数据结构大小也并不惊人。同时，cluster map的状态更新并不会频繁发生。即便如此，Ceph依然对cluster map信息的扩散机制进行了优化，以便减轻相关计算和通信压力：**首先**，cluster map信息是以增量形式扩散的。如果任意一次通信的双方发现其epoch不一致，则版本更新的一方将把二者所拥有的cluster map的差异发送给另外一方。**其次**，cluster map信息是以异步且lazy的形式扩散的。也即，monitor并不会在每一次cluster map版本更新后都将新版本广播至全体OSD，而是在有OSD向自己上报信息时，将更新回复给对方。类似的，各个OSD也是在和其他OSD通信时，将更新发送给版本低于自己的对方。

基于上述机制，Ceph避免了由于cluster map版本更新而引起的广播风暴。这虽然是一种异步且lazy的机制，但对于一个由n个OSD组成的Ceph集群，任何一次版本更新能够在O(log(n))时间复杂度内扩散到集群中的任何一个OSD上。

**一个可能被问到的问题是：**既然这是一种异步和lazy的扩散机制，则在版本扩散过程中，系统必定出现各个OSD看到的cluster map不一致的情况，这是否会导致问题？答案是：不会。事实上，如果一个client和它要访问的PG内部的各个OSD看到的cluster map状态一致，则访问操作就可以正确进行。而如果这个client或者PG中的某个OSD和其他几方的cluster map不一致，则根据Ceph的机制设计，这几方将首先同步cluster map至最新状态，并进行必要的数据re-balancing操作，然后即可继续正常访问。

**2. Ceph基本组件之说明**



如上图所示，**Ceph主要有三个基本进程：**  
**<1> Osd**: 用于集群中所有数据与对象的存储。处理集群数据的复制、恢复、回填、再均衡。并向其他osd守护进程发送心跳，然后向Mon提供一些监控信息。当Ceph存储集群设定数据有两个副本时（一共存两份），则至少需要两个OSD守护进程即两个OSD节点，集群才能达到active+clean状态。  
**<2> MDS**(可选)：为Ceph文件系统提供元数据计算、缓存与同步。在ceph中，元数据也是存储在osd节点中的，mds类似于元数据的代理缓存服务器。MDS进程并不是必须的进程，只有需要使用CEPHFS时，才需要配置MDS节点。  
**<3> Monitor**：监控整个集群Cluster map的状态，维护集群的cluster MAP二进制表，保证集群数据的一致性。ClusterMAP描述了对象块存储的物理位置，以及一个将设备聚合到物理位置的桶列表。

通常来说，一块磁盘和该磁盘对应的守护进程称为一个OSD。守护进程的作用是从该磁盘读取和写入数据。该磁盘可以是一个硬盘或者SSD盘或者RAID0，总之是一个逻辑磁盘。如果一个节点只有一个守护进程和对应的磁盘，那么该OSD就成了一个节点。通常一个节点有多个OSD守护进程和多个磁盘，所以通常来说OSD不是一个节点。

**Ceph要求必须是奇数个Monitor监控节点，一般建议至少是3个**（如果是自己私下测试玩玩的话，可以是1个，但是生产环境绝不建议1个）用于维护和监控整个集群的状态，每个Monitor都有一个Cluster Map，只要有这个Map，就能够清楚知道每个对象存储在什么位置了。客户端会先tcp连接到Monitor，从中获取Cluster Map，并在客户端进行计算，当知道对象的位置后，再直接与OSD通信（去中心化的思想）。OSD节点平常会向Monitor节点发送简单心跳，只有当添加、删除或者出现异常状况时，才会自动上报信息给Monitor。

**MDS是可选的，只有需要使用Ceph FS的时候才需要配置MDS节点**。在Ceph中，元数据也是存放在OSD中的，MDS只相当于元数据的缓存服务器。

**在Ceph中，如果要写数据，只能向主OSD写，然后再由主OSD向从OSD同步地写，只有当从OSD返回结果给主OSD后，主OSD才会向客户端报告写入完成的消息**。如果要读数据，不会使用读写分离，而是也需要先向主OSD发请求，以保证数据的强一致性。

**三、Ceph分布式存储集群简单部署  [一个ceph集群至少需要3个OSD才能实现冗余和高可用性，并且至少需要一个mon和一个msd。其中mon和msd可以部署在其中一个osd节点上 ]**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549  550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697 | 0）基本信息：  这里我只是测试环境，所以使用一个监控节点，三个存储节点，具体如下：    ip地址              主机名           ceph磁盘          备注  192.168.10.200      ceph-node1       20G               作为mds、mon、osd0  192.168.10.201      ceph-node2       20G               作为osd1  192.168.10.202      ceph-node3       20G               作为osd2  192.168.10.203      ceph-client      挂载点：/cephfs   ceph客户端    Ceph的文件系统作为一个目录挂载到客户端cephclient的/cephfs目录下，可以像操作普通目录一样对此目录进行操作。    1）安装前准备  分别在ceph的三个节点机（ceph-node1、ceph-node2、ceph-node3）上添加hosts  [root@ceph-node1 ~]# cat /etc/hosts  127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4  ::1         localhost localhost.localdomain localhost6 localhost6.localdomain6  192.168.10.200 ceph-node1  192.168.10.201 ceph-node2  192.168.10.202 ceph-node3    添加完hosts后，做下测试，保证使用hosts中映射的主机名能ping通。  [root@ceph-node3 ~]# ping ceph-node1  PING ceph-node1 (192.168.10.200) 56(84) bytes of data.  64 bytes from ceph-node1 (192.168.10.200): icmp\_seq=1 ttl=64 time=0.211 ms  64 bytes from ceph-node1 (192.168.10.200): icmp\_seq=2 ttl=64 time=0.177 ms    [root@ceph-node3 ~]# ping ceph-node2  PING ceph-node2 (192.168.10.201) 56(84) bytes of data.  64 bytes from ceph-node2 (192.168.10.201): icmp\_seq=1 ttl=64 time=1.27 ms  64 bytes from ceph-node2 (192.168.10.201): icmp\_seq=2 ttl=64 time=0.169 ms    分别在ceph的三个节点机（ceph-node1、ceph-node2、ceph-node3）上创建用户ceph，密码统一设置为ceph  [root@ceph-node1 ~]# adduser ceph  [root@ceph-node1 ~]# echo "ceph"|passwd --stdin ceph  Changing password for user ceph.  passwd: all authentication tokens updated successfully.    在每个Ceph节点中为用户增加 root 权限  [root@ceph-node1 ~]# echo "ceph ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/ceph  [root@ceph-node1 ~]# chmod 0440 /etc/sudoers.d/ceph    测试是否具有sudo权限  [root@ceph-node1 ~]# su - ceph  [ceph@ceph-node1 ~]$ sudo su -  [root@ceph-node1 ~]#    关闭防火墙  [root@ceph-node1 ~]# service iptables stop  [root@ceph-node1 ~]# chkconfig iptables off  [root@ceph-node1 ~]# setenforce 0  [root@ceph-node1 ~]# getenforce  Permissive  [root@ceph-node1 ~]# sed -i 's\_SELINUX=enforcing\_SELINUX=disabled\_g' /etc/sysconfig/selinux    2）Ceph管理节点部署设置（root用户操作）。这个只需要在ceph-node1监控节点操作即可。  增加Ceph资料库至 ceph-deploy 管理节点，之后安装 ceph-deploy。  [root@ceph-node1 ~]# rpm -Uvh http://download.ceph.com/rpm-hammer/el6/noarch/ceph-release-1-1.el6.noarch.rpm  [root@ceph-node1 ~]# yum install ceph-deploy -y    3）数据节点磁盘挂载（root用户）  在ceph-node1、ceph-node2、ceph-node3上分别挂载了一块20G大小的磁盘作为ceph的数据存储测试使用。  需要将它们分区，创建xfs文件系统。    由于本案例中的四个虚拟机均是使用WebvirtMgr创建的kvm虚拟机  创建kvm虚拟机，具体参考：http://www.cnblogs.com/kevingrace/p/5737724.html    现在需要在宿主机上创建3个20G的盘，然后将这三个盘分别挂载到ceph-node1、ceph-node2、ceph-node3这三个虚拟机上  操作记录如下：    在宿主机上的操作  [root@kvm-server ~]# virsh list --all   Id    Name                           State  ----------------------------------------------------   1     ceph-node1                     running   2     ceph-node2                     running   3     ceph-node3                     running   4     centos6-04                     running    [root@zabbix-server ~]# ll /data/kvm/ios/\*     #这里的/data/kvm/ios是webvirtmgr里创建虚拟机时定义的镜像存放地址  total 31751800  -rw-r--r--. 1 qemu qemu 3972005888 Jan 16 17:13 CentOS-6.9-x86\_64-bin-DVD1.iso  -rw-r--r--. 1 root root 4521459712 Jan 16 17:14 CentOS-7-x86\_64-DVD-1708.iso  -rw-------. 1 qemu qemu 4914610176 Feb  6 17:58 ceph-node1.img  -rw-------. 1 qemu qemu 4417716224 Feb  6 17:59 ceph-node2.img  -rw-------. 1 qemu qemu 4405723136 Feb  6 17:58 ceph-node4.img  -rw-------. 1 qemu qemu 3420389376 Feb  6 17:58 disk004.img    创建一个20G的新磁盘挂载到ceph-node1虚拟机上  [root@zabbix-server ~]# qemu-img create -f raw /data/kvm/ios/ceph01.img 20G  Formatting '/data/kvm/ios/ceph01.img', fmt=raw size=21474836480  [root@zabbix-server ~]# ll /data/kvm/ios/ceph01.img  -rw-r--r--. 1 root root 21474836480 Feb  6 18:00 /data/kvm/ios/ceph01.img  [root@zabbix-server ~]# du -sh /data/kvm/ios/ceph01.img  0  /data/kvm/ios/ceph01.img  [root@zabbix-server ~]# virsh attach-disk centos6-01 /data/kvm/ios/ceph01.img vdb --cache none          # 如果卸载，命令为：virsh detach-disk centos6-01 /data/kvm/ios/ceph01.img  Disk attached successfully    然后到ceph-node1虚拟机上查看，发现新建立的20G的磁盘/dev/vdb已经挂载上来了  [root@ceph-node1 ~]# fdisk -l  .........    Disk /dev/vdb: 21.5 GB, 21474836480 bytes  16 heads, 63 sectors/track, 41610 cylinders  Units = cylinders of 1008 \* 512 = 516096 bytes  Sector size (logical/physical): 512 bytes / 512 bytes  I/O size (minimum/optimal): 512 bytes / 512 bytes  Disk identifier: 0x00000000    在虚拟机上对新挂载的磁盘进行创建xfs文件系统操作  [root@ceph-node1 ~]# parted /dev/vdb  GNU Parted 2.1  Using /dev/vdb  Welcome to GNU Parted! Type 'help' to view a list of commands.  (parted) mklabel gpt                                                  //在此输入mklabel gpt  Warning: The existing disk label on /dev/vdb will be destroyed and all data on this disk will be lost. Do you want to continue?  Yes/No? yes                                                           //在此输入yes  (parted) mkpart primary xfs 0% 100%                                   //在此输入mkpart primary xfs 0% 100%  (parted) quit                                                         //在此输入quit  Information: You may need to update /etc/fstab.    [root@ceph-node1 ~]# fdisk -l  .......  Disk /dev/vdb: 21.5 GB, 21474836480 bytes  255 heads, 63 sectors/track, 2610 cylinders  Units = cylinders of 16065 \* 512 = 8225280 bytes  Sector size (logical/physical): 512 bytes / 512 bytes  I/O size (minimum/optimal): 512 bytes / 512 bytes  Disk identifier: 0x00000000       Device Boot      Start         End      Blocks   Id  System  /dev/vdb1               1        2611    20971519+  ee  GPT    [root@ceph-node1 ~]# mkfs.xfs /dev/vdb1  meta-data=/dev/vdb1              isize=256    agcount=4, agsize=1310592 blks           =                       sectsz=512   attr=2, projid32bit=0  data     =                       bsize=4096   blocks=5242368, imaxpct=25           =                       sunit=0      swidth=0 blks  naming   =version 2              bsize=4096   ascii-ci=0  log      =internal log           bsize=4096   blocks=2560, version=2           =                       sectsz=512   sunit=0 blks, lazy-count=1  realtime =none                   extsz=4096   blocks=0, rtextents=0  ————————————————————————————————————————————————————————————————————————————————————————————  报错：mkfs.xfs error: command not found.  解决办法：yum install -y xfsprogs kmod-xfs  ————————————————————————————————————————————————————————————————————————————————————————————    另外两台虚拟机ceph-node2和ceph-node3挂载新磁盘以及创建xfs文件系统的方法和上面一样，这里不做赘录了。    4）存储集群搭建（ceph用户）  集群规模：1个管理节点(即安装ceph-deploy的节点)，1个监控节点，3个数据节点。    ---------------首先完成ceph-node1、ceph-node2、ceph-node3三个节点机在ceph用户下的ssh无密码登陆的信任关系（ssh无密码操作过程省略）---------------  做完ceph用户之间的ssh信任关系后，可以简单测试下。  [ceph@ceph-node1 ~]$ ssh -p22 ceph@ceph-node2  [ceph@ceph-node1 ~]$ ssh -p22 ceph@ceph-node3    [ceph@ceph-node2 ~]$ ssh -p22 ceph@ceph-node1  [ceph@ceph-node2 ~]$ ssh -p22 ceph@ceph-node3    [ceph@ceph-node3 ~]$ ssh -p22 ceph@ceph-node1  [ceph@ceph-node3 ~]$ ssh -p22 ceph@ceph-node2    ---------------接着在管理节点上使用ceph-deploy（管理节点上使用ceph-deploy，所以下面操作都要在ceph-node1机器上完成）---------------  [ceph@ceph-node1 ~]$ mkdir ceph-cluster  [ceph@ceph-node1 ~]$ cd ceph-cluster  [ceph@ceph-node1 ceph-cluster]$    创建一个ceph新集群（注意下面命令中ceph-node1是monit监控节点的主机名），设置ceph-node1为mon节点  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy new ceph-node1  [ceph\_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf  [ceph\_deploy.cli][INFO  ] Invoked (1.5.37): /usr/bin/ceph-deploy new ceph-node1  [ceph\_deploy.cli][INFO  ] ceph-deploy options:  [ceph\_deploy.cli][INFO  ]  username                      : None  [ceph\_deploy.cli][INFO  ]  func                          : <function new at 0x112b500>  [ceph\_deploy.cli][INFO  ]  verbose                       : False  [ceph\_deploy.cli][INFO  ]  overwrite\_conf                : False  [ceph\_deploy.cli][INFO  ]  quiet                         : False  [ceph\_deploy.cli][INFO  ]  cd\_conf                       : <ceph\_deploy.conf.cephdeploy.Conf instance at 0xf26e18>  [ceph\_deploy.cli][INFO  ]  cluster                       : ceph  [ceph\_deploy.cli][INFO  ]  ssh\_copykey                   : True  [ceph\_deploy.cli][INFO  ]  mon                           : ['ceph-node1']  [ceph\_deploy.cli][INFO  ]  public\_network                : None  [ceph\_deploy.cli][INFO  ]  ceph\_conf                     : None  [ceph\_deploy.cli][INFO  ]  cluster\_network               : None  [ceph\_deploy.cli][INFO  ]  default\_release               : False  [ceph\_deploy.cli][INFO  ]  fsid                          : None  [ceph\_deploy.new][DEBUG ] Creating new cluster named ceph  [ceph\_deploy.new][INFO  ] making sure passwordless SSH succeeds  [ceph-node1][DEBUG ] connection detected need for sudo  [ceph-node1][DEBUG ] connected to host: ceph-node1  [ceph-node1][DEBUG ] detect platform information from remote host  [ceph-node1][DEBUG ] detect machine type  [ceph-node1][DEBUG ] find the location of an executable  [ceph-node1][INFO  ] Running command: sudo /sbin/ip link show  [ceph-node1][INFO  ] Running command: sudo /sbin/ip addr show  [ceph-node1][DEBUG ] IP addresses found: [u'192.168.10.200']  [ceph\_deploy.new][DEBUG ] Resolving host ceph-node1  [ceph\_deploy.new][DEBUG ] Monitor ceph-node1 at 192.168.10.200  [ceph\_deploy.new][DEBUG ] Monitor initial members are ['ceph-node1']  [ceph\_deploy.new][DEBUG ] Monitor addrs are ['192.168.10.200']  [ceph\_deploy.new][DEBUG ] Creating a random mon key...  [ceph\_deploy.new][DEBUG ] Writing monitor keyring to ceph.mon.keyring...  [ceph\_deploy.new][DEBUG ] Writing initial config to ceph.conf...  Error in sys.exitfunc:    [ceph@ceph-node1 ceph-cluster]$ ll  total 28  -rw-rw-r--. 1 ceph ceph   202 Feb  6 10:37 ceph.conf  -rw-rw-r--. 1 ceph ceph 16774 Feb  6 10:55 ceph-deploy-ceph.log  -rw-------. 1 ceph ceph    73 Feb  6 10:37 ceph.mon.keyring    ---------------安装ceph。这个需要在3台存储节点机器上都要安装ceph（root用户操作）---------------  [ceph@ceph-node1 ~]# yum install -y yum-plugin-prioritie  [ceph@ceph-node1 ~]# rpm -Uvh http://download.ceph.com/rpm-hammer/el6/noarch/ceph-release-1-1.el6.noarch.rpm  [ceph@ceph-node1 ~]# yum -y install ceph    [ceph@ceph-node2 ~]# yum install -y yum-plugin-prioritie  [ceph@ceph-node2 ~]# rpm -Uvh http://download.ceph.com/rpm-hammer/el6/noarch/ceph-release-1-1.el6.noarch.rpm  [ceph@ceph-node2 ~]# yum -y install ceph    [ceph@ceph-node3 ~]# yum install -y yum-plugin-prioritie  [ceph@ceph-node3 ~]# rpm -Uvh http://download.ceph.com/rpm-hammer/el6/noarch/ceph-release-1-1.el6.noarch.rpm  [ceph@ceph-node3 ~]# yum -y install ceph  ————————————————————————————————————————————————————————————————————————————————————————————————  报错报错：  安装ceph报错：  warning: rpmts\_HdrFromFdno: Header V4 RSA/SHA1 Signature, key ID 460f3994: NOKEY  Retrieving key from https://download.ceph.com/keys/release.asc  Importing GPG key 0x460F3994:   Userid: "Ceph.com (release key) <security@ceph.com>"   From  : https://download.ceph.com/keys/release.asc  warning: rpmts\_HdrFromFdno: Header V3 RSA/SHA256 Signature, key ID 0608b895: NOKEY    Public key for libunwind-1.1-3.el6.x86\_64.rpm is not installed    解决办法：  Centos6.x系统下的处理办法：  yum install -y yum-utils && yum-config-manager --add-repo https://dl.fedoraproject.org/pub/epel/6/x86\_64/ && yum install --nogpgcheck -y epel-release && rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6 && rm -f /etc/yum.repos.d/dl.fedoraproject.org\*    Centos7.x系统下的处理办法：  yum install -y yum-utils && yum-config-manager --add-repo https://dl.fedoraproject.org/pub/epel/7/x86\_64/ && yum install --nogpgcheck -y epel-release && rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7 && rm -f /etc/yum.repos.d/dl.fedoraproject.org\*  ——————————————————————————————————————————————————————————————————————————————————————————————    ---------------接着创建并初始化监控节点（ceph-deploy是管理节点上的操作，下面操作都要在ceph-node1机器上完成）---------------  [ceph@ceph-node1 ~]$ cd ceph-cluster/  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy mon create-initial  .......  .......  [ceph-node1][INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph --admin-daemon=/var/run/ceph/ceph-mon.ceph-node1.asok mon\_status  [ceph-node1][INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph --name mon. --keyring=/var/lib/ceph/mon/ceph-ceph-node1/keyring auth get client.admin  [ceph-node1][INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph --name mon. --keyring=/var/lib/ceph/mon/ceph-ceph-node1/keyring auth get client.bootstrap-mds  [ceph-node1][INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph --name mon. --keyring=/var/lib/ceph/mon/ceph-ceph-node1/keyring auth get client.bootstrap-osd  [ceph-node1][INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph --name mon. --keyring=/var/lib/ceph/mon/ceph-ceph-node1/keyring auth get client.bootstrap-rgw  [ceph\_deploy.gatherkeys][INFO  ] Storing ceph.client.admin.keyring  [ceph\_deploy.gatherkeys][INFO  ] Storing ceph.bootstrap-mds.keyring  [ceph\_deploy.gatherkeys][INFO  ] keyring 'ceph.mon.keyring' already exists  [ceph\_deploy.gatherkeys][INFO  ] Storing ceph.bootstrap-osd.keyring  [ceph\_deploy.gatherkeys][INFO  ] Storing ceph.bootstrap-rgw.keyring  [ceph\_deploy.gatherkeys][INFO  ] Destroy temp directory /tmp/tmp8sI0aU  Error in sys.exitfunc:    ————————————————————————————————————————————————————————————————————————————————————————————————————  如果出现报错（如果有WARNIN告警信息，则不用理会，不影响结果）:  [ERROR] Failed to execute command: /usr/sbin/service ceph -c /etc/ceph/ceph.conf start mon.ceph01  [ERROR] GenericError: Failed to create 1 monitors    解决办法如下：  [ceph@ceph-node1 ~]$ sudo yum install redhat-lsb  ————————————————————————————————————————————————————————————————————————————————————————————————————    执行此命令后会初始化mon结点，并且在mon结点生成ceph.conf等文件，ceph.conf文件中声明了mon结点等信息  [ceph@ceph-node1 ceph-cluster]$ ll  total 76  -rw-------. 1 ceph ceph   113 Feb  6 11:23 ceph.bootstrap-mds.keyring  -rw-------. 1 ceph ceph   113 Feb  6 11:23 ceph.bootstrap-osd.keyring  -rw-------. 1 ceph ceph   113 Feb  6 11:23 ceph.bootstrap-rgw.keyring  -rw-------. 1 ceph ceph   127 Feb  6 11:23 ceph.client.admin.keyring  -rw-rw-r--. 1 ceph ceph   202 Feb  6 10:37 ceph.conf  -rw-rw-r--. 1 ceph ceph 47659 Feb  6 11:23 ceph-deploy-ceph.log  -rw-------. 1 ceph ceph    73 Feb  6 10:37 ceph.mon.keyring  [ceph@ceph-node1 ceph-cluster]$ cat ceph.conf  [global]  fsid = 12d4f209-69ea-4ad9-9507-b00557b42815  mon\_initial\_members = ceph-node1  mon\_host = 192.168.10.200  auth\_cluster\_required = cephx  auth\_service\_required = cephx  auth\_client\_required = cephx    查看一下Ceph存储节点的硬盘情况：  [ceph@ceph-node1 ceph-cluster]$ pwd  /home/ceph/ceph-cluster  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy disk list ceph-node1  [ceph\_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf  [ceph\_deploy.cli][INFO  ] Invoked (1.5.37): /usr/bin/ceph-deploy disk list ceph-node1  [ceph\_deploy.cli][INFO  ] ceph-deploy options:  [ceph\_deploy.cli][INFO  ]  username                      : None  [ceph\_deploy.cli][INFO  ]  verbose                       : False  [ceph\_deploy.cli][INFO  ]  overwrite\_conf                : False  [ceph\_deploy.cli][INFO  ]  subcommand                    : list  [ceph\_deploy.cli][INFO  ]  quiet                         : False  [ceph\_deploy.cli][INFO  ]  cd\_conf                       : <ceph\_deploy.conf.cephdeploy.Conf instance at 0x18af8c0>  [ceph\_deploy.cli][INFO  ]  cluster                       : ceph  [ceph\_deploy.cli][INFO  ]  func                          : <function disk at 0x18a2488>  [ceph\_deploy.cli][INFO  ]  ceph\_conf                     : None  [ceph\_deploy.cli][INFO  ]  default\_release               : False  [ceph\_deploy.cli][INFO  ]  disk                          : [('ceph-node1', None, None)]  [ceph-node1][DEBUG ] connection detected need for sudo  [ceph-node1][DEBUG ] connected to host: ceph-node1  [ceph-node1][DEBUG ] detect platform information from remote host  [ceph-node1][DEBUG ] detect machine type  [ceph-node1][DEBUG ] find the location of an executable  [ceph\_deploy.osd][INFO  ] Distro info: CentOS 6.9 Final  [ceph\_deploy.osd][DEBUG ] Listing disks on ceph-node1...  [ceph-node1][DEBUG ] find the location of an executable  [ceph-node1][INFO  ] Running command: sudo /usr/sbin/ceph-disk list  [ceph-node1][WARNIN] WARNING:ceph-disk:Old blkid does not support ID\_PART\_ENTRY\_\* fields, trying sgdisk; may not correctly identify ceph volumes with dmcrypt  [ceph-node1][DEBUG ] /dev/sr0 other, iso9660  [ceph-node1][DEBUG ] /dev/vda :  [ceph-node1][DEBUG ]  /dev/vda1 other, ext4, mounted on /boot  [ceph-node1][DEBUG ]  /dev/vda2 other, LVM2\_member  [ceph-node1][DEBUG ] /dev/vdb :  [ceph-node1][DEBUG ]  /dev/vdb1 other, xfs  Error in sys.exitfunc:    同理查看其它两个ceph节点的硬盘情况  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy disk list ceph-node2  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy disk list ceph-node3    ——————————————————————————————————————————————————————————————————————————————————————————————————  如果报错：  [ERROR ] RuntimeError: remote connection got closed, ensure ``requiretty`` is disabled for ceph02    解决办法：  需要在ceph-node1、ceph-node2、ceph-node3三个osd节点中设置sudo权限（root账号下），如下：  [root@ceph-node1 ~]# visudo  .......  # Defaults requiretty                     //如有有一行内容，就注释掉  Defaults:ceph !requiretty                 //添加这一行内容  ——————————————————————————————————————————————————————————————————————————————————————————————————    ---------------添加数据节点（在三个osd节点的root账号下操作）---------------  [root@ceph-node1 ~]# mkdir /data  [root@ceph-node1 ~]# mkdir /data/osd0  [root@ceph-node1 ~]# chmod -R 777 /data/    [root@ceph-node2 ~]# mkdir /data  [root@ceph-node2 ~]# mkdir /data/osd1  [root@ceph-node2 ~]# chmod -R 777 /data/    [root@ceph-node3 ~]# mkdir /data  [root@ceph-node3 ~]# mkdir /data/osd2  [root@ceph-node3 ~]# chmod -R 777 /data/    ---------------挂载ceph磁盘---------------  在三个osd节点上将上面各个新建的ceph磁盘分别挂载到对应的目录  [root@ceph-node1 ~]# mount /dev/vdb1 /data/osd0    [root@ceph-node2 ~]# mount /dev/vdb1 /data/osd1    [root@ceph-node3 ~]# mount /dev/vdb1 /data/osd2    --------------在管理节点准备OSD（本案例中的管理节点就是ceph-node1节点），在ceph用户下操作--------------  [ceph@ceph-node1 ~]$ cd ceph-cluster/  [ceph@ceph-node1 ceph-cluster]$ pwd  /home/ceph/ceph-cluster  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy osd prepare ceph-node1:/data/osd0 ceph-node2:/data/osd1 ceph-node3:/data/osd2  .......  .......  [ceph-node3][INFO  ] checking OSD status...  [ceph-node3][DEBUG ] find the location of an executable  [ceph-node3][INFO  ] Running command: sudo /usr/bin/ceph --cluster=ceph osd stat --format=json  [ceph\_deploy.osd][DEBUG ] Host ceph-node3 is now ready for osd use.  Error in sys.exitfunc:    ---------------激活OSD（仍然在管理节点ceph-node1上操作）-----------------  （注意如果osd的目录所在磁盘是ext4格式的文件系统会报错，需要进行额外的设置）  [ceph@ceph-node1 ceph-cluster]$ pwd  /home/ceph/ceph-cluster  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy osd activate ceph-node1:/data/osd0 ceph-node2:/data/osd1 ceph-node3:/data/osd2  .......  .......  [ceph-node3][INFO  ] checking OSD status...  [ceph-node3][DEBUG ] find the location of an executable  [ceph-node3][INFO  ] Running command: sudo /usr/bin/ceph --cluster=ceph osd stat --format=json  [ceph-node3][INFO  ] Running command: sudo chkconfig ceph on  Error in sys.exitfunc:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  如果报错：  [WARNIN] ceph disk: Error: No cluster conf fonud in /etc/ceph with fsid dobbb980-a7a0-4562-ab8b-8d423424234  [ERROR] RuntimeError: command returned non-zero exit status: 1  [ERROR] RuntimeError: Failed to execute command: ceph-disk -v activate --mark-init sysvinit --mount /osd0    解决办法：  这是是因为多次卸载和安装ceph造成磁盘的id和cluster的uuid不一致，需要将ceph-node1的/data/osd0下的文件全部清空  （每台osd节点机对应目录下的文件都要情况，即ceph-node2的/data/osd1下的文件、ceph-node3的/data/osd2下的文件）  ——————————————————————————————————————————————————————————————————————————————————————————————————————————————    ------------------开机挂载磁盘（三台osd节点都要操作）-------------------  挂载好的新磁盘会在下一次重启或开机的时候失效，因此需要在每台机上对新磁盘进行挂载，并编辑/etc/fstab文件  [root@ceph-node1 ~]# cat /etc/fstab  ........  /dev/vdb1               /data/osd0                   xfs     defaults        0 0    [root@ceph-node2 ~]# cat /etc/fstab  ........  /dev/vdb1               /data/osd1                   xfs     defaults        0 0    [root@ceph-node3 ~]# cat /etc/fstab  ........  /dev/vdb1               /data/osd2                   xfs     defaults        0 0    ----------------------------分发配置和密钥----------------------------  使用ceph-deploy命令将配置文件和管理密钥复制到管理节点和它的Ceph节点。  [ceph@ceph-node1 ~]$ cd ceph-cluster/  [ceph@ceph-node1 ceph-cluster]$ pwd  /home/ceph/ceph-cluster  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy admin ceph-node1 ceph-node2 ceph-node3  [ceph\_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf  [ceph\_deploy.cli][INFO  ] Invoked (1.5.37): /usr/bin/ceph-deploy admin ceph-node1 ceph-node2 ceph-node3  [ceph\_deploy.cli][INFO  ] ceph-deploy options:  [ceph\_deploy.cli][INFO  ]  username                      : None  [ceph\_deploy.cli][INFO  ]  verbose                       : False  [ceph\_deploy.cli][INFO  ]  overwrite\_conf                : False  [ceph\_deploy.cli][INFO  ]  quiet                         : False  [ceph\_deploy.cli][INFO  ]  cd\_conf                       : <ceph\_deploy.conf.cephdeploy.Conf instance at 0x230ca70>  [ceph\_deploy.cli][INFO  ]  cluster                       : ceph  [ceph\_deploy.cli][INFO  ]  client                        : ['ceph-node1', 'ceph-node2', 'ceph-node3']  [ceph\_deploy.cli][INFO  ]  func                          : <function admin at 0x2263050>  [ceph\_deploy.cli][INFO  ]  ceph\_conf                     : None  [ceph\_deploy.cli][INFO  ]  default\_release               : False  [ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph-node1  [ceph-node1][DEBUG ] connection detected need for sudo  [ceph-node1][DEBUG ] connected to host: ceph-node1  [ceph-node1][DEBUG ] detect platform information from remote host  [ceph-node1][DEBUG ] detect machine type  [ceph-node1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf  [ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph-node2  [ceph-node2][DEBUG ] connection detected need for sudo  [ceph-node2][DEBUG ] connected to host: ceph-node2  [ceph-node2][DEBUG ] detect platform information from remote host  [ceph-node2][DEBUG ] detect machine type  [ceph-node2][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf  [ceph\_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph-node3  [ceph-node3][DEBUG ] connection detected need for sudo  [ceph-node3][DEBUG ] connected to host: ceph-node3  [ceph-node3][DEBUG ] detect platform information from remote host  [ceph-node3][DEBUG ] detect machine type  [ceph-node3][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf  Error in sys.exitfunc:    -----------------------------------查看集群状态（ceph用户或root用户下都可以查看）------------------------------  [ceph@ceph-node1 ~]$ ceph health  HEALTH\_OK  [ceph@ceph-node1 ~]$ ceph -s      cluster 12d4f209-69ea-4ad9-9507-b00557b42815       health HEALTH\_OK       monmap e1: 1 mons at {ceph-node1=192.168.10.200:6789/0}              election epoch 2, quorum 0 ceph-node1       osdmap e14: 3 osds: 3 up, 3 in        pgmap v26: 64 pgs, 1 pools, 0 bytes data, 0 objects              15459 MB used, 45944 MB / 61404 MB avail                    64 active+clean    出现上面信息就表示ceph分布式存储集群环境已经成功了！！  ——————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————  ======如果报错1：  2018-02-07 21:53:54.886552 7fdc5cf5a700 -1 monclient(hunting): ERROR: missing keyring, cannot use cephx for authentication  2016-02-07 21:53:54.886584 7fdc5cf5a700  0 librados: client.admin initialization error (2) No such file or directory  Error connecting to cluster: ObjectNotFound    是因为普通用户无法读取导致无法进行cephx认证，执行以下命令：  sudo chmod +r /etc/ceph/ceph.client.admin.keyring    ======如果报错2：  HEALTH\_WARN 64 pgs degraded; 64 pgs stuck degraded; 64 pgs stuck unclean; 64 pgs stuck undersized; 64 pgs undersized    是因为配置中默认osd=3，备份=2，与实际不符，更改配置文件ceph.conf，增加以下两行：  osd\_pool\_default\_size = 2  osd\_pool\_default\_min\_size = 1  （不过没有生效，怀疑需要重启集群，待测试），或者添加一个osd节点（有些说法是要最少3个osd）    ======如果报错3：  2018-02-07 15:21:05.597951 7fde9cb6b700 -1 monclient(hunting): ERROR: missing keyring, cannot use cephx for authentication  2018-02-07 15:21:05.597963 7fde9cb6b700  0 librados: client.admin initialization error (2) No such file or directory    是因为没有权限读取keyring，执行以下命令赋权限：  sudo chmod 777 /etc/ceph/\*  ——————————————————————————————————————————————————————————————————————————————————————————————————————————————————————————    通过上面操作步骤，三个ods节点的ceph服务默认是启动着的。  [root@ceph-node1 ~]# /etc/init.d/ceph status  === mon.ceph-node1 ===  mon.ceph-node1: running {"version":"0.94.10"}  === osd.0 ===  osd.0: running {"version":"0.94.10"}    [root@ceph-node2 ~]# /etc/init.d/ceph status  === osd.1 ===  osd.1: running {"version":"0.94.10"}    [root@ceph-node3 ~]# /etc/init.d/ceph status  === osd.2 ===  osd.2: running {"version":"0.94.10"}    ========================================重新部署（温馨提示）==========================================  部署过程中如果出现任何奇怪的问题无法解决，可以简单的删除所有节点(即把所有的mon节点、osd节点、mds节点全部删除)，一切从头再来：  # ceph-deploy purge ceph-node1 ceph-node2 ceph-node3  # ceph-deploy purgedata ceph-node1 ceph-node2 ceph-node3  # ceph-deploy forgetkey  ======================================================================================================    5）创建文件系统  创建文件系统的步骤参考官网：http://docs.ceph.com/docs/master/cephfs/createfs/  对于一个刚创建的MDS服务，虽然服务是运行的，但是它的状态直到创建 pools 以及文件系统的时候才会变为Active.    还没有创建时候的状态  [ceph@ceph-node1 ~]$ ceph mds stat  e1: 0/0/0 up    -------------------创建管理节点（ceph-node1节点上）-------------------  [ceph@ceph-node1 ~]$ cd ceph-cluster/  [ceph@ceph-node1 ceph-cluster]$ pwd  /home/ceph/ceph-cluster  [ceph@ceph-node1 ceph-cluster]$ ceph-deploy mds create ceph-node1  [ceph\_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf  [ceph\_deploy.cli][INFO  ] Invoked (1.5.37): /usr/bin/ceph-deploy mds create ceph-node1  [ceph\_deploy.cli][INFO  ] ceph-deploy options:  [ceph\_deploy.cli][INFO  ]  username                      : None  [ceph\_deploy.cli][INFO  ]  verbose                       : False  [ceph\_deploy.cli][INFO  ]  overwrite\_conf                : False  [ceph\_deploy.cli][INFO  ]  subcommand                    : create  [ceph\_deploy.cli][INFO  ]  quiet                         : False  [ceph\_deploy.cli][INFO  ]  cd\_conf                       : <ceph\_deploy.conf.cephdeploy.Conf instance at 0xcf0290>  [ceph\_deploy.cli][INFO  ]  cluster                       : ceph  [ceph\_deploy.cli][INFO  ]  func                          : <function mds at 0xcd8398>  [ceph\_deploy.cli][INFO  ]  ceph\_conf                     : None  [ceph\_deploy.cli][INFO  ]  mds                           : [('ceph-node1', 'ceph-node1')]  [ceph\_deploy.cli][INFO  ]  default\_release               : False  [ceph\_deploy.mds][DEBUG ] Deploying mds, cluster ceph hosts ceph-node1:ceph-node1  [ceph-node1][DEBUG ] connection detected need for sudo  [ceph-node1][DEBUG ] connected to host: ceph-node1  [ceph-node1][DEBUG ] detect platform information from remote host  [ceph-node1][DEBUG ] detect machine type  [ceph\_deploy.mds][INFO  ] Distro info: CentOS 6.9 Final  [ceph\_deploy.mds][DEBUG ] remote host will use sysvinit  [ceph\_deploy.mds][DEBUG ] deploying mds bootstrap to ceph-node1  [ceph-node1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf  [ceph-node1][DEBUG ] create path if it doesn't exist  [ceph-node1][INFO  ] Running command: sudo ceph --cluster ceph --name client.bootstrap-mds --keyring /var/lib/ceph/bootstrap-mds/ceph.keyring auth get-or-create mds.ceph-node1 osd allow rwx mds allow mon allow profile mds -o /var/lib/ceph/mds/ceph-ceph-node1/keyring  [ceph-node1][INFO  ] Running command: sudo service ceph start mds.ceph-node1  [ceph-node1][DEBUG ] === mds.ceph-node1 ===  [ceph-node1][DEBUG ] Starting Ceph mds.ceph-node1 on ceph-node1...  [ceph-node1][DEBUG ] starting mds.ceph-node1 at :/0  [ceph-node1][INFO  ] Running command: sudo chkconfig ceph on  Error in sys.exitfunc:    注意：如果不创建mds管理节点，client客户端将不能正常挂载到ceph集群！！    [ceph@ceph-node1 ceph-cluster]$ sudo /etc/init.d/ceph status  === mon.ceph-node1 ===  mon.ceph-node1: running {"version":"0.94.10"}  === osd.0 ===  osd.0: running {"version":"0.94.10"}  === mds.ceph-node1 ===  mds.ceph-node1: running {"version":"0.94.10"}    [ceph@ceph-node1 ceph-cluster]$ ceph mds stat  e1: 0/0/0 up    --------------------创建pool------------------  查看pool。pool是ceph存储数据时的逻辑分区,它起到namespace的作用  [ceph@ceph-node1 ceph-cluster]$ ceph osd lspools  0 rbd,    新创建的ceph集群只有rdb一个pool。这时需要创建一个新的pool  新建pool，可参考官网：http://docs.ceph.com/docs/master/rados/operations/pools/ <br>  [ceph@ceph-node1 ceph-cluster]$ ceph osd pool create cephfs\_data 10                //后面的数字是PG的数量  pool 'cephfs\_data' created  [ceph@ceph-node1 ceph-cluster]$ ceph osd pool create cephfs\_metadata 10            //创建pool的元数据  pool 'cephfs\_metadata' created  [ceph@ceph-node1 ceph-cluster]$ ceph fs new myceph  cephfs\_metadata cephfs\_data  new fs with metadata pool 2 and data pool 1    再次查看pool  [ceph@ceph-node1 ceph-cluster]$ ceph osd lspools  0 rbd,1 cephfs\_data,2 cephfs\_metadata,    --------------------检验--------------------  [ceph@ceph-node1 ~]$ ceph mds stat  e5: 1/1/1 up {0=ceph-node1=up:active}    查看集群状态  [ceph@ceph-node1 ~]$ ceph -s      cluster 12d4f209-69ea-4ad9-9507-b00557b42815       health HEALTH\_OK       monmap e1: 1 mons at {ceph-node1=192.168.10.200:6789/0}              election epoch 2, quorum 0 ceph-node1       mdsmap e5: 1/1/1 up {0=ceph-node1=up:active}       osdmap e19: 3 osds: 3 up, 3 in        pgmap v37: 84 pgs, 3 pools, 1962 bytes data, 20 objects              15460 MB used, 45943 MB / 61404 MB avail                    84 active+clean    查看ceph集群端口  [root@ceph-node1 ~]# lsof -i:6789  COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME  ceph-mon  568 root   13u  IPv4  65149      0t0  TCP ceph-node1:smc-https (LISTEN)  ceph-mon  568 root   23u  IPv4  73785      0t0  TCP ceph-node1:smc-https->ceph-node1:34280 (ESTABLISHED)  ceph-mon  568 root   24u  IPv4  73874      0t0  TCP ceph-node1:smc-https->ceph-node2:42094 (ESTABLISHED)  ceph-mon  568 root   25u  IPv4  73920      0t0  TCP ceph-node1:smc-https->ceph-node3:42080 (ESTABLISHED)  ceph-mon  568 root   26u  IPv4  76829      0t0  TCP ceph-node1:smc-https->ceph-node1:34354 (ESTABLISHED)  ceph-osd 4566 root   24u  IPv4  73784      0t0  TCP ceph-node1:34280->ceph-node1:smc-https (ESTABLISHED)  ceph-mds 7928 root    9u  IPv4  76828      0t0  TCP ceph-node1:34354->ceph-node1:smc-https (ESTABLISHED)    6）client挂载  client挂载方式有两种：内核kernal方式和fuse方式。由于我们使用的centos6.7内核版本太低，而升级内核时间太长，故直接采用fuse方式进行挂载。    特别需要注意的是：  client挂载前需要在client节点安装ceph-fuse，方法参考上面的操作即可！    安装ceph-fuse  [root@ceph-client ~]# rpm -Uvh https://dl.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm  [root@ceph-client ~]# yum install -y ceph-fuse    创建挂载目录  [root@ceph-client ~]# mkdir /cephfs    复制配置文件  将ceph配置文件ceph.conf从管理节点copy到client节点  [root@ceph-client ~]# rsync -e "ssh -p22" -avp root@192.168.10.200:/etc/ceph/ceph.conf /etc/ceph/  [root@ceph-client ~]# ll /etc/ceph/ceph.conf  -rw-r--r--. 1 root root 202 Feb  7 03:32 /etc/ceph/ceph.conf    复制密钥  将ceph的ceph.client.admin.keyring从管理节点copy到client节点  [root@ceph-client ~]# rsync -e "ssh -p22" -avp root@192.168.10.200:/etc/ceph/ceph.client.admin.keyring /etc/ceph/  [root@ceph-client ~]# ll /etc/ceph/ceph.client.admin.keyring  -rwxrwxrwx. 1 root root 127 Feb  7 02:55 /etc/ceph/ceph.client.admin.keyring    查看ceph授权  [root@ceph-client ~]# ceph auth list  installed auth entries:    mds.ceph-node1     key: AQDV/3lasqfVARAAa/eOiCbWQ/ccFLkVD/9UCQ==     caps: [mds] allow     caps: [mon] allow profile mds     caps: [osd] allow rwx  osd.0     key: AQAB9XlapzMfOxAAfKLo3Z6FZMyqWUW9F5FU4Q==     caps: [mon] allow profile osd     caps: [osd] allow \*  osd.1     key: AQAL9XlaT06zLRAABQr9kweLPiHL2Icdgj8YsA==     caps: [mon] allow profile osd     caps: [osd] allow \*  osd.2     key: AQAV9XladRLjORAATbxZHpD34ztqJJ3vFGkVOg==     caps: [mon] allow profile osd     caps: [osd] allow \*  client.admin     key: AQCnH3lalDbeHhAAC6y7YERqDDGL+f8S+sP1bw==     caps: [mds] allow     caps: [mon] allow \*     caps: [osd] allow \*  client.bootstrap-mds     key: AQCoH3laUqT5JRAAiVMRF8ueWQXfgQjIPalNMQ==     caps: [mon] allow profile bootstrap-mds  client.bootstrap-osd     key: AQCnH3laBzU7MBAAnreE1I/0Egh8gvMseoy93w==     caps: [mon] allow profile bootstrap-osd  client.bootstrap-rgw     key: AQCoH3laziwXDxAAenU6cqw0gRcQ3Y0JtygtPQ==     caps: [mon] allow profile bootstrap-rgw      挂载到指定目录  将ceph挂载到/cephfs  [root@ceph-client ~]# ceph-fuse -m 192.168.10.200:6789 /cephfs  ceph-fuse[10466]: starting ceph client  2018-02-07 04:08:47.227156 7fb6afd9e760 -1 init, newargv = 0x33d23e0 newargc=11  ceph-fuse[10466]: starting fuse    [root@ceph-client ~]# df -h  Filesystem      Size  Used Avail Use% Mounted on  /dev/vda2        48G  1.5G   45G   4% /  tmpfs           1.9G     0  1.9G   0% /dev/shm  /dev/vda1       190M   36M  145M  20% /boot  /dev/vda5        47G   52M   44G   1% /home  ceph-fuse        60G   16G   45G  26% /cephfs    如上可以看到，clinet节点已经将三个osd存储节点（每个20G）的ceph存储挂载上了，总共60G！  可以在/cephfs下存放数据，当ceph的部分osd节点挂掉后，不会影响client对ceph存储数据的读取操作！    取消挂载  [root@ceph-client ~]# umount /cephfs |

**如果client是ubuntu系统，则使用fuse挂载ceph存储的操作方法**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | 安装ceph-fuse  # apt-get install -y ceph-fuse    或者手动安装  下载地址：http://mirrors.aliyun.com/ubuntu/pool/main/c/ceph/  下载包ceph-dbg\_0.94.1-0ubuntu1\_amd64.deb  # dpkg -i ceph-dbg\_0.94.1-0ubuntu1\_amd64.deb    创建挂载目录  # mkdir /cephfs    复制配置文件  将ceph配置文件ceph.conf从管理节点copy到client节点  # rsync -e "ssh -p22" -avp root@192.168.10.200:/etc/ceph/ceph.conf /etc/ceph/    复制密钥  将ceph的ceph.client.admin.keyring从管理节点copy到client节点  # rsync -e "ssh -p22" -avp root@192.168.10.200:/etc/ceph/ceph.client.admin.keyring /etc/ceph/    查看ceph授权  # ceph auth list    挂载到指定目录  将ceph挂载到/cephfs  # ceph-fuse -m 192.168.10.200:6789 /cephfs    取消挂载  # umount /cephfs |

当有一半以上的OSD节点挂掉后，远程客户端挂载的Ceph存储就会使用异常了，即暂停使用。比如本案例中有3个OSD节点，当其中一个OSD节点挂掉后，客户端挂载的Ceph存储使用正常；但当有2个OSD节点挂掉后，客户端挂载的Ceph存储就不能正常使用了（表现为Ceph存储目录下的数据读写操作一直卡着状态），当OSD节点恢复后，Ceph存储也会恢复正常使用。

**四、Ceph日常操作命令**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420 | 查看状态命令：  查看ceph集群状态：ceph -s  查看mon状态：ceph mon stat  查看msd状态：ceph msd stat  查看osd状态：ceph osd stat  查看osd目录树（可以查看每个osd挂在哪台机，是否已启动）：ceph osd tree    启动ceph进程命令：  需要在对应的节点进行启动（如果对应节点没有该服务，会进行提示）  启动mon进程：service ceph start mon.ceph-node1  启动msd进程：service ceph start msd.ceoh-node1  启动osd进程：service ceph start osd.0（在ceph-node1上）  启动osd进程：service ceph start osd.1（在ceph-node2上）  启动osd进程：service ceph start osd.2（在ceph-node3上）    查看机器的监控状态  # ceph health    查看ceph的实时运行状态  # ceph -w    检查信息状态信息  # ceph -s    查看ceph存储空间  [root@client ~]# ceph df    删除一个节点的所有的ceph数据包  # ceph-deploy purge ceph-node1  # ceph-deploy purgedata ceph-node1    为ceph创建一个admin用户并为admin用户创建一个密钥，把密钥保存到/etc/ceph目录下：  # ceph auth get-or-create client.admin mds 'allow' osd 'allow ' mon 'allow ' > /etc/ceph/ceph.client.admin.keyring  或  # ceph auth get-or-create client.admin mds 'allow' osd 'allow ' mon 'allow ' -o /etc/ceph/ceph.client.admin.keyring    为osd.0创建一个用户并创建一个key  # ceph auth get-or-create osd.0 mon 'allow rwx' osd 'allow \*' -o /var/lib/ceph/osd/ceph-0/keyring    为mds.node1创建一个用户并创建一个key（ceph-node1是节点名称）  # ceph auth get-or-create mds.node1 mon 'allow rwx' osd 'allow ' mds 'allow ' -o /var/lib/ceph/mds/ceph-ceph-node1/keyring    查看ceph集群中的认证用户及相关的key  ceph auth list    删除集群中的一个认证用户  ceph auth del osd.0    查看集群的详细配置（ceph-node1是节点名称）  # ceph daemon mon.ceph-node1 config show | more    查看集群健康状态细节  # ceph health detail    查看ceph log日志所在的目录  # ceph-conf –name mon.node1 –show-config-value log\_file    =================关于mon节点的相关操作命令====================  查看mon的状态信息  # ceph mon stat    查看mon的选举状态  # ceph quorum\_status    查看mon的映射信息  # ceph mon dump    删除一个mon节点  # ceph mon remove node1    获得一个正在运行的mon map，并保存在1.txt文件中  # ceph mon getmap -o 1.txt    查看上面获得的map  # monmaptool --print 1.txt    map注入新加入的节点（如新节点主机名为ceph-node4）  # ceph-mon -i ceph-node4 --inject-monmap 1.txt    查看mon的amin socket  # ceph-conf --name mon.ceph-node1 --show-config-value admin\_socket    查看mon的详细状态（ceph-node1为mon节点主机名）  # ceph daemon mon.ceph-node1 mon\_status    删除一个mon节点（ceph-node1为mon节点主机名）  # ceph mon remove ceph-node1    =================msd节点相关操作命令====================  查看msd状态  # ceph mds stat    查看msd的映射信息  # ceph mds dump    删除一个mds节点  # ceph mds rm 0 mds.ceph-node1    =================osd节点相关操作命令====================  查看ceph osd运行状态  # ceph osd stat    查看osd映射信息  # ceph osd dump    查看osd的目录树  # ceph osd tree    down掉一个osd硬盘(比如down掉osd.0节点磁盘）  # ceph osd down 0    在集群中删除一个osd硬盘  # ceph osd rm 0    在集群中删除一个osd 硬盘 crush map  # ceph osd crush rm osd.0    在集群中删除一个osd的host节点  # ceph osd crush rm node1    查看最大osd的个数（默认最大是4个osd节点）  # ceph osd getmaxosd    设置最大的osd的个数（当扩大osd节点的时候必须扩大这个值）  # ceph osd setmaxosd 10    设置osd crush的权重为1.0  ceph osd crush set {id} {weight} [{loc1} [{loc2} …]]  例如：  [root@ceph-node1 ~]# ceph osd crush set 3 3.0 host=ceph-node4  set item id 3 name 'osd.3' weight 3 at location {host=node4} to crush map  [root@ceph-node1 ~]# ceph osd tree  # id weight type name up/down reweight  -1 6 root default  -2 1 host ceph-node1  0 1 osd.0 up 1  -3 1 host ceph-node2  1 1 osd.1 up 1  -4 1 host ceph-node3  2 1 osd.2 up 1  -5 3 host ceph-node4  3 3 osd.3 up 0.5    或者用下面的方式  [root@ceph-node1 ~]# ceph osd crush reweight osd.3 1.0  reweighted item id 3 name 'osd.3' to 1 in crush map  [root@ceph-node1 ~]# ceph osd tree  # id weight type name up/down reweight  -1 4 root default  -2 1 host node1  0 1 osd.0 up 1  -3 1 host node2  1 1 osd.1 up 1  -4 1 host node3  2 1 osd.2 up 1  -5 1 host node4  3 1 osd.3 up 0.5      设置osd的权重    [root@ceph-node1 ~]# ceph osd reweight 3 0.5  reweighted osd.3 to 0.5 (8327682)  [root@ceph-node1 ~]# ceph osd tree  # id weight type name up/down reweight  -1 4 root default  -2 1 host node1  0 1 osd.0 up 1  -3 1 host node2  1 1 osd.1 up 1  -4 1 host node3  2 1 osd.2 up 1  -5 1 host node4  3 1 osd.3 up 0.5    把一个osd节点逐出集群    [root@ceph-node1 ~]# ceph osd out osd.3  marked out osd.3.  [root@ceph-node1 ~]# ceph osd tree  # id weight type name up/down reweight  -1 4 root default  -2 1 host node1  0 1 osd.0 up 1  -3 1 host node2  1 1 osd.1 up 1  -4 1 host node3  2 1 osd.2 up 1  -5 1 host node4  3 1 osd.3 up 0  # osd.3的reweight变为0了就不再分配数据，但是设备还是存活的    把逐出的osd加入集群    [root@ceph-node1 ~]# ceph osd in osd.3  marked in osd.3.  [root@ceph-node1 ~]# ceph osd tree  # id weight type name up/down reweight  -1 4 root default  -2 1 host node1  0 1 osd.0 up 1  -3 1 host node2  1 1 osd.1 up 1  -4 1 host node3  2 1 osd.2 up 1  -5 1 host node4  3 1 osd.3 up 1    暂停osd （暂停后整个集群不再接收数据）  [root@ceph-node1 ~]# ceph osd pause  # set pauserd,pausewr    再次开启osd （开启后再次接收数据）  [root@ceph-node1 ~]# ceph osd unpause  # unset pauserd,pausewr    查看一个集群osd.2参数的配置  # ceph –admin-daemon /var/run/ceph/ceph-osd.2.asok config show | less    ======================PG组相关操作命令=======================  查看pg组的映射信息  # ceph pg dump    查看一个PG的map    [root@client ~]# ceph pg map 0.3f  osdmap e88 pg 0.3f (0.3f) -> up [0,2] acting [0,2]  #其中的[0,2]代表存储在osd.0、osd.2节点，osd.0代表主副本的存储位置    查看PG状态  [root@client ~]# ceph pg stat  v1164: 448 pgs: 448 active+clean; 10003 MB data, 23617 MB used, 37792 MB / 61410 MB avail    查询一个pg的详细信息  [root@client ~]# ceph pg 0.26 query    查看pg中stuck的状态  [root@client ~]# ceph pg dump\_stuck unclean  ok  [root@client ~]# ceph pg dump\_stuck inactive  ok  [root@client ~]# ceph pg dump\_stuck stale  ok    显示一个集群中的所有的pg统计  # ceph pg dump –format plain    恢复一个丢失的pg  # ceph pg {pg-id} mark\_unfound\_lost revert    显示非正常状态的pg  # ceph pg dump\_stuck inactive|unclean|stale    ======================pool相关操作命令========================  查看ceph集群中的pool数量  [root@ceph-node1 ~]# ceph osd lspools  0 data,1 metadata,2 rbd,    在ceph集群中创建一个pool  # ceph osd pool create kevin 100  这里的100指的是PG组，kevin是集群名称    为一个ceph pool配置配额  # ceph osd pool set-quota data max\_objects 10000    在集群中删除一个pool（集群名字需要重复两次）  # ceph osd pool delete kevin kevin --yes-i-really-really-mean-it      显示集群中pool的详细信息  [root@ceph-node1 ~]# rados df  pool name category KB objects clones degraded unfound rd rd KB wr wr KB  data - 475764704 116155 0 0 0 0 0 116379 475764704  metadata - 5606 21 0 0 0 0 0 314 5833  rbd - 0 0 0 0 0 0 0 0 0  total used 955852448 116176  total avail 639497596  total space 1595350044  [root@ceph-node1 ~]#    给一个pool创建一个快照  [root@ceph-node1 ~]# ceph osd pool mksnap data date-snap  created pool data snap date-snap    删除pool的快照  [root@ceph-node1 ~]# ceph osd pool rmsnap data date-snap  removed pool data snap date-snap    查看data池的pg数量  [root@ceph-node1 ~]# ceph osd pool get data pg\_num  pg\_num: 64    设置data池的最大存储空间为100T（默认是1T)  [root@ceph-node1 ~]# ceph osd pool set data target\_max\_bytes 100000000000000  set pool 0 target\_max\_bytes to 100000000000000    设置data池的副本数是3  [root@ceph-node1 ~]# ceph osd pool set data size 3  set pool 0 size to 3    设置data池能接受写操作的最小副本为2  [root@ceph-node1 ~]# ceph osd pool set data min\_size 2  set pool 0 min\_size to 2    查看集群中所有pool的副本尺寸  [root@admin mycephfs]# ceph osd dump | grep 'replicated size'  pool 0 'data' replicated size 3 min\_size 2 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 26 owner 0 flags hashpspool crash\_replay\_interval 45 target\_bytes 100000000000000 stripe\_width 0  pool 1 'metadata' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool stripe\_width 0  pool 2 'rbd' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool stripe\_width 0    设置一个pool的pg数量  [root@ceph-node1 ~]# ceph osd pool set data pg\_num 100  set pool 0 pg\_num to 100    设置一个pool的pgp数量  [root@ceph-node1 ~]# ceph osd pool set data pgp\_num 100  set pool 0 pgp\_num to 100    =================================rados指令====================================  查看ceph集群中有多少个pool （只是查看pool)  [root@ceph-node1 ~]# rados lspools  rbd  cephfs\_data  cephfs\_metadata    查看ceph集群中有多少个pool,并且每个pool容量及利用情况  [root@ceph-node1 ~]# rados df  pool name                 KB      objects       clones     degraded      unfound           rd        rd KB           wr        wr KB  cephfs\_data                0            0            0            0           0            0            0            0            0  cephfs\_metadata           21           20            0            0           0            0            0           45           36  rbd                        0            0            0            0           0            0            0            0            0    total used        15833064           20    total avail       47044632    total space       62877696    创建一个pool,名称为kevin  [root@ceph-node1 ~]# rados mkpool kevin  successfully created pool kevin    查看ceph pool中的ceph object （这里的object是以块形式存储的）  [root@ceph-node1 ~]# rados ls -p kevin|more    创建一个对象object（下面的kevin是pool名称）  [root@ceph-node1 ~]# rados create kevin-object -p kevin  [root@ceph-node1 ~]# rados -p kevin ls  kevin-object    删除一个对象  [root@ceph-node1 ~]# rados rm kevin-object -p kevin  [root@ceph-node1 ~]# rados -p kevin ls    =================================rbd命令的用法=====================================  查看ceph中一个pool里的所有镜像  # rbd ls kevin     //kevin是一个pool名  或者  # rbd list kevin    查看ceph pool中一个镜像的信息（kevin是pool名，wangshibo是镜像名）  [root@ceph ~]# rbd info -p kevin --image wangshibo    在kevin池中创建一个命名为wangshibo的10000M的镜像  [root@ceph-node1 ~]# rbd create -p kevin --size 10000 wangshibo  [root@ceph-node1 ~]# rbd -p kevin info wangshibo                 //查看新建的镜像的信息  rbd image 'wangshibo':     size 10000 MB in 2500 objects     order 22 (4096 kB objects)     block\_name\_prefix: rb.0.1079.2ae8944a     format: 1    删除一个镜像  [root@ceph-node1 ~]# rbd rm -p kevin wangshibo  Removing image: 100% complete...done.    调整一个镜像的尺寸（前提是wangshibo镜像已经创建并没有被删除）  [root@ceph-node1 ~]# rbd resize -p kevin --size 20000 wangshibo  Resizing image: 100% complete...done.    查看调整后的wangshibo镜像大小  [root@ceph-node1 ~]# rbd -p kevin info wangshibo  rbd image 'wangshibo':     size 20000 MB in 5000 objects     order 22 (4096 kB objects)     block\_name\_prefix: rb.0.107d.2ae8944a     format: 1    给一个镜像创建一个快照（如下，池/镜像@快照 ）  [root@ceph-node1 ~]# rbd snap create kevin/wangshibo@wangshibo123    查看快照  [root@ceph-node1 ~]# rbd info kevin/wangshibo@wangshibo123  rbd image 'wangshibo':     size 20000 MB in 5000 objects     order 22 (4096 kB objects)     block\_name\_prefix: rb.0.107d.2ae8944a     format: 1     protected: False    查看一个镜像文件的快照  [root@ceph-node1 ~]# rbd snap ls -p kevin wangshibo  SNAPID NAME             SIZE       4 wangshibo123 20000 MB    删除一个镜像文件的一个快照  [root@ceph-node1 ~]# rbd snap rm kevin/wangshibo@wangshibo123  [root@ceph-node1 ~]# rbd snap ls -p kevin wangshibo        //wangshibo123快照已经被删除    如果发现不能删除显示的报错信息是此快照备写保护了，下面命令是删除写保护后再进行删除。  # rbd snap unprotect kevin/wangshibo@wangshibo123  # rbd snap rm kevin/wangshibo@wangshibo123    删除一个镜像文件的所有快照  [root@ceph-node1 ~]# rbd snap purge -p kevin wangshibo  Removing all snapshots: 100% complete...done.    把ceph pool中的一个镜像导出  [root@ceph-node1 ~]# rbd export -p kevin --image wangshibo  Exporting image: 100% complete...done.    把一个镜像导入ceph中 （但是直接导入是不能用的，因为没有经过openstack,openstack是看不到的）  [root@ceph-node1 ~]# rbd import /root/ceph\_test.img -p kevin --image wangshibo |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*