誉天教育
YU TIAN EDU

Linux云计算架构师涨薪班

Ceph存储池

学习目标

- 存储池介绍
- 存储池类型
- 复制池
- 纠删码池
- 纠删码规则
- 存储池配额与快照
- 存储池命名空间
- 存储池参数管理

存储池介绍

- 池是ceph存储集群的逻辑分区，用于存储对象
- 池具有特定的属性
  - 池类型：一种用于提供数据冗余，一种用于提升数据的存取效率
  - PG数量：将对象存储到由CRUSH算法决定的一组OSD中
  - 访问级别：不同用户的访问权限
- 对象存储到池中时，使用CRUSH规则将该对象分配到池中的一个PG，PG根据池的配置和CRUSH算法自动映射到一组OSD
- 池中PG数量对性能有重要影响。通常而言，池应当配置为每个OSD包含100-200个归置组
- 创建池时，ceph会检查每个OSD的PG数是否超过200。如果超过，ceph不会创建这个池。（200的限制可通过mon_max_pg_per_osd参数修改）

存储池类型

- 用于提供数据冗余
  - 复制池
  - 纠删码池
- 用于提升数据读写效率
  - 缓存池

复制池

ceph osd pool create <pool-name> [pg-num] [pgp-num] [replicated]

- pool-name 存储池的名称
- pg-num 存储池的pg总数
- pgp-num 存储池的pg的有效数，通常与pg相等（通俗理解PG存放的OSD排列组合）
- replicated 指定为复制池，即使不指定，默认也是创建复制池

pg的数量影响性能（需要将pg从一个osd移动到另一个osd）：

pg数量过多: 数据移动时，每个PG维护的数据量过少，ceph占用大量的cpu和内存计算，影响集群正常客户端使用

pg数量过少:单个pg存储的数据就越多，移动pg会占用大量带宽，影响集群客户端使用

为池启用ceph应用

- 创建池后，必须显式指定能够使用它的ceph应用类型：
  - ceph块设备 ceph对象网关 ceph文件系统
- 如果不显示指定类型，集群将显示HEALTH_WARN状态（使用ceph health detail命令查看）
- 为池关联应用类型：ceph osd pool application enable pool-name app
  - cephfs
  - rbd
  - rgw
- 示例：
  - ceph osd pool application enable myfirstpool rbd

查询池信息

- 列出存储池
  - ceph osd pool ls 查询池
  - ceph osd pool ls detail 查询池的详细信息
- 获取池统计信息
  - ceph df：获取池用量统计数据
  - ceph osd df：获取osd上磁盘使用量统计数据
  - ceph osd pool stats：获取池性能统计数据

存储池对象操作

- 上传对象
  - rados -p 池名 put 对象 上传的文件
- 下载对象
  - rados -p 池名 get 对象 下载的文件
- 删除对象
  - rados -p 池名 rm 对象
- 查看对象所在的PG
  - ceph osd map 池名 对象名
- 查询pg的主OSD
  - ceph pg dump pgs_brief

存储池配额

- 配额方式
  - 对象配额　　容量配额
- 语法提示:
  - ceph osd pool set-quota pool-name max_objects obj-count max_bytes bytes
- 示例
  - ceph osd pool set-quota myfirstpool max_objects 1000
- 可将值设置为0来删除配额。同时通过ceph　df命令查看池的用量统计数据
- 当ceph达到池配额时，操作会被无限期阻止

重命名存储池

- 语法
  - ceph osd pool rename current-name new-name
- 示例
  - ceph osd pool rename mysecondpool mytestpool
- 重命名池，不影响池中的数据

创建存储池快照

- 创建快照
  - ceph osd pool mksnap pool-name snap-name
- 查看快照
  - rados -p pool-name lssnap
  - rados -p pool-name -s snap-name ls
- 删除快照
  - ceph osd pool rmsnap pool-name snap-name
- 回滚快照
  - rados -p pool-name -s snap-name get object-name file 下载对象到本地
  - rados -p pool-name rollback object-name snap-name 还原对象到存储池

配置存储池参数

- 设置池参数
  - ceph osd pool set pool-name parameter value
- 获取池参数
  - ceph osd pool get pool-name parameter
- 列出所有参数及其值
  - ceph osd pool get pool-name all

PG的计算方法

$$\frac{(\text{每个 OSD 的目标 PG}) * (\text{OSD \#}) * (\%\text{数据})}{(\text{大小})}$$

Target PGs per OSD：预估每个OSD的PG数，一般取100计算
集群OSD不增加推荐值为 100 增加为 200
OSD #：集群OSD数量
%Data：预估该存储池占该OSD集群总容量的近似百分比
Size：该存储池的副本数
最终池的PG数取值： 取最接近2的n次幂结果，与最近结果低于25%则使用下一个幂的结果

- If the result of this calculation is less than (OSD#)/(Size), then the PG Count is updated to (OSD#)/(Size). This tactic ensures an even load/data distribution by allocating at least one Primary or Secondary PG to every OSD for every Pool.
- The output value of 1 above is then rounded to the nearest value of 2. This rounding marginally improves the efficiency of the CRUSH algorithm.
- If the nearest power of 2 is more than 25% below the original value (the result of the first equation), we use the next higher power of 2

PG推荐值

- 一种比较通用的取值规则：
  - 少于5个OSD时可把pg_num设置为128
  - OSD数量在5到10个时，可把pg_num设置为512
  - OSD数量在10到50个时，可把pg_num设置为4096
  - OSD数量大于50时，建议自行计算
- PG计算器
  - pgcalc：https://ceph.com/pgcalc （ceph官方已废弃）
  - cephpgc：https://access.redhat.com/labs/cephpgc

PG autoscaler横空出世

- PG支持分裂
  - 现有PG可以将其内容“拆分”为许多较小的PG，从而增加了池中PG的总数
- PG支持合并
  - 现有两个PG“合并”到一个更大的PG中，从而减少池中PG的总数（N版以后开始支持）

ceph mgr module enable pg_autoscaler N版以后默认开启

集群根据每个池中实际存储（或预期要存储）的数据量，并自动选择适当的pg_num值

pg和pgp

- PG存储池中的个数，PGP是存储池PG的OSD排列组合数
- 扩容pg → pg中的对象会进行移动 在新的osd上生成pg
- 扩容pgp-> pg中的对象不会进行移动 会引起部分pg在osd上的分布

在池中配置命名空间

- namespace是池中对象的逻辑组。可以限制用户对池的访问，使得用户只能存储或检索这个namespace内的对象
- namesapce的优点是能够使将用户访问权限池的某一部分
- namespace目前仅支持使用librados的应用，不支持rgw和rbd
- 若要在命名空间内存储对象，客户端应用必须提供池和命名空间的名称
- 默认情况下，每个池包含一个具有空名称的namespace，称为默认namespace
- rados命令可以通过-N name或者--namespace=name选项存储和检索池中指定命名空间的对象
- 示例
  - rados -p mytestpool -N system put srv /etc/services
  - rados -p mytestpool -N system ls
  - rados -p mytestpool --all ls
  - rados -p mytestpool --all ls --format=json | python3 -m json.tool
    - --all 可列出池中所有命名空间中的所有对象
    - --format=json 返回json格式的结果

删除存储池

- 删除池
  - ceph osd pool delete pool-name pool-name --yes-i-really-really-mean-it
- 从L版开始，已将mon_allow_pool_delete配置参数设置为false，以提供额外的保护。即使借助--yes-i-really-really-mean-it选项，ceph osd pool delete命令也不会导致池被删除
- 可以将mon_allow_pool_delete参数设置为true，然后重启mon服务，以允许删除池
- 即使mon_allow_pool_delete被设置为true，也可以通过在池级别上将nodelete选项设置为true来防止池被删除：
  - ceph osd pool set pool-name nodelete true

纠删码池

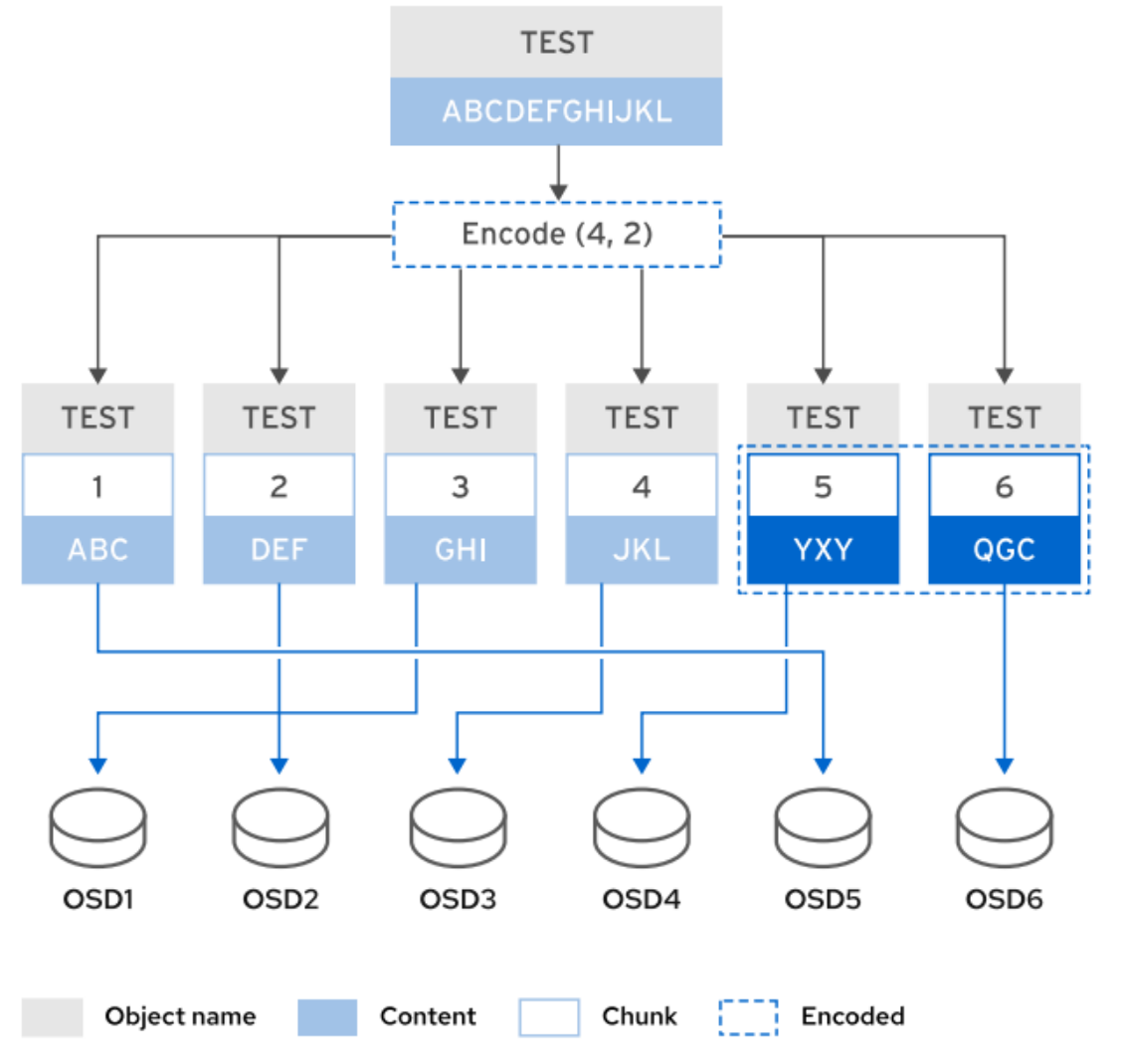
- 纠删码池使用纠删码而非复制来保护对象数据
- 相对于复制池，纠删码池会节约存储空间，但是需要更多的计算资源
- 纠删码池一般只能用于对象存储（从L版开始可以调整为支持rbd和cephfs）
- 打开池选项allow_ec_overwrites以支持rbd和cephfs
- 纠删码池L版之前不支持快照

纠删码原理

- 纠删码的存储方法是将每个object划分成更小的数据块，每一个数据块称为data chunk，再用编码块（coding chunk）对它们进行编码，最后将这些数据块和编码块存储到Ceph集群的不同故障域中，从而保证数据安全。纠删码概念的核心公式n=k+m，解释如下：
  - k：原始object被划分成的数据块的个数
  - m：附加到所有原始数据块的额外编码块的个数
  - n：执行纠删码处理后，所创建的块的总数

  n=K+M

纠删码池工作原理



- 与复制相比，纠删代码使用存储容量的效率更高。复制池维护对象的 n 个副本，而纠删代码仅维护 k + m 个区块。例如，具有 3 个副本的复制池要使用 3 倍存储空间。而 k=4 和 m=2 的纠删代码池仅 要使用 1.5 倍存储空间

创建纠删码池

- 语法
  - ceph osd pool create <pool-name> <pg-num> [pgp-num] erasure [erasure-code-profile] [cursh-ruleset-name] [expected_num_objects]
  - erasure用于指定创建一个纠删码池
  - erasure-code-profile是要使用的profile的名称，可以使用ceph osd erasure-code-profile set命令创建新的profile。profile定义使用的插件类型以及k和m的值。默认情况下，ceph使用default profile

纠删码profiles

example: ceph osd erasure-code-profile set ecdemo k=2 m=1 crush-failure-domain=osd

- k：在不同 OSD 之间拆分的数据区块数量。默认值为 2。
- m：数据变得不可用之前可以出现故障的 OSD 数量。默认值为 1。
- plugin: 此可选参数定义要使用的纠删代码算法。
- crush-failure-domain：CRUSH 故障域，默认设置为 host
- crush-device-class: 典型的类别可能包括 hdd、ssd 或nvme。
- crush-root：此可选参数设置 CRUSH 规则集的根节点。
- key=value: 插件可以具有对该插件唯一的键值参数。
- technique：每个插件提供一组不同的技术来实施不同的算法。

红帽推荐纠删码配置:
4+2（比率为 1:1.5）
8+3（比率为 1:1.375）
8+4（比率为 1:1.5）

纠删码profiles

- crush-device-class：指定设备类别。在 Ceph 中，设备可以按照不同的类别进行分类，如 ssd、hdd 等。这个参数用于指定当前存储池所使用的设备类别。
- crush-failure-domain：指定故障域级别。在 Ceph 中，可以将存储设备划分为不同的故障域，以便更好地处理设备故障。例如，将设备分组成主机、机架或数据中心等。
- crush-root：指定 CRUSH 映射中的根名称。CRUSH 是一个 Ceph 集群的数据分布算法，它将数据和元数据映射到存储设备上。这个参数指定了 CRUSH 映射中的根名称。
- jerasure-per-chunk-alignment：指定 Jerasure 编码库是否启用块对齐方式。Jerasure 是 Ceph 存储系统中的一个编码库，用于提供纠删码编码和解码功能。这个参数用于配置 Jerasure 是否启用块对齐。
- k：指定数据块数量。在本例中，k=3，表示原始数据被分成了 3 份，并生成了 2 个纠删码块。
- m：指定纠删码块数量。在本例中，m=2，表示生成了 2 个纠删码块。
- plugin：指定使用的编码库。在本例中，使用的是 Jerasure 编码库。
- technique：指定纠删码编码技术。在本例中，使用的是 Reed-Solomon-Vandermonde 纠删码编码技术。
- w：指定字长。在 Jerasure 编码库中，字长指的是每个编码块中的数据块数量。在本例中，w=8，表示每个编码块包含 8 个数据块

纠删码profiles管理

- 使用指定profile创建纠删码池
  - ceph osd pool create pool-name erasure profile-name
- 列出现有的配置
  - ceph osd erasure-code-profile ls
- 删除现有的配置
  - ceph osd erasure-code-profile rm profile-name
- 查看现有的配置
  - ceph osd erasure-code-profile get profile-name

PS：无法修改或更新现有纠删码池的profile

誉天教育
YU TIAN EDU

Thank you