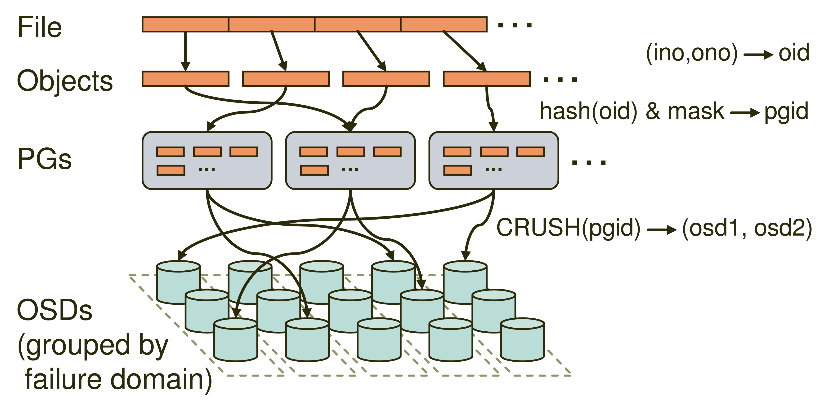
简单总结一下之前看的有关文件的读写过程。

### 读写过程



上图大体可以表示从文件到保存在存储实体上的过程，详细步骤如下：

1. RADOS中需要配置Object Size的值，也就是每个Object大小的最大值，一般情况下会设置为2M到4M。当一个file存进ceph中时，会根据Object Size将file进行切分。若file的大小不足Object Size，则直接按原大小映射到一个Object中；若file的大小大于Object Size，则file会被切分为若干个完整大小的Object，和一个不完整大小的Object。对于小文件，映射不足一个完整的Object时，并不会发生填充，Object大小就是原始文件的大小。

Object的大小，可以在ceph.file.layout中object\_size进行配置。

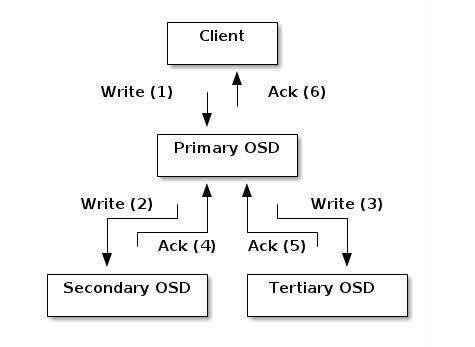
$ touch file

$ getfattr -n ceph.file.layout file

# file: file

ceph.file.layout="stripe\_unit=4194304 stripe\_count=1 object\_size=4194304 pool=cephfs\_data"

1. 从Object映射到PG的过程就是简单的Hash运算。将Object id计算出一个hash值然后与mask位与就可以得到PG的id值。
2. Crush算法的作用是定位数据存放的位置，在这个步骤中就是将PG映射到一个OSD set中。在三副本replicated下，set中就会有三个osd。第一个就是primary osd，负责接受读写请求和维护一致性以及进行修复。后两个就是replicated osd。
3. 写文件的过程是，client先写到primary osd上，然后再由primary osd写到replicated osd上。为了维护强一致性，具体的流程如下。

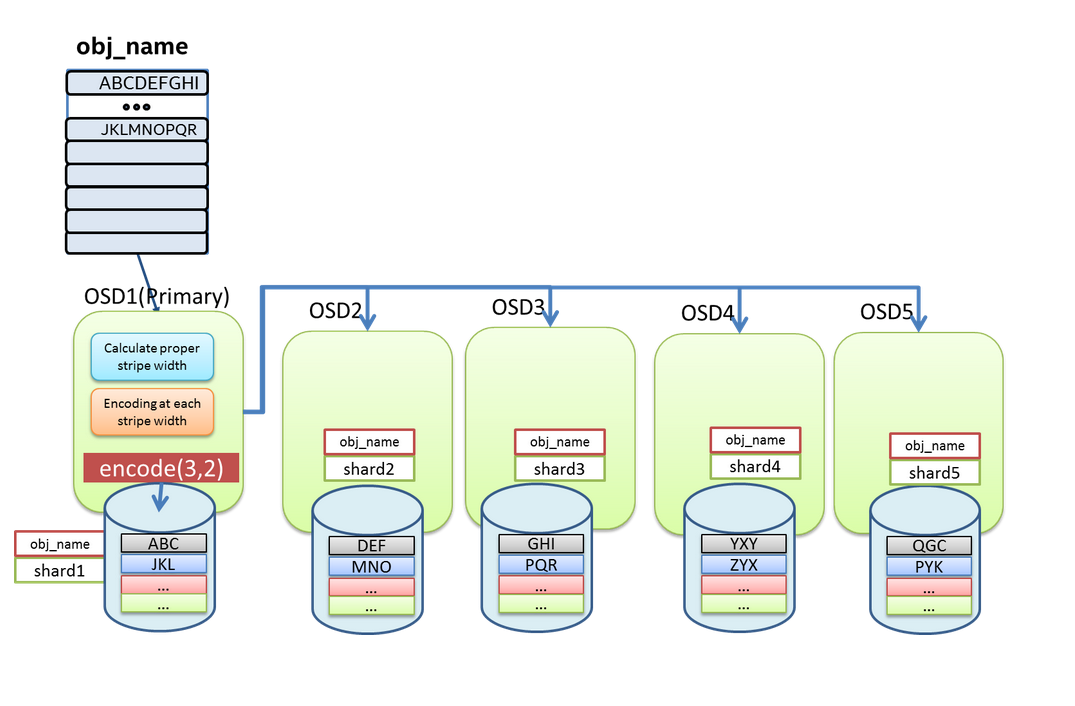


1. 通过上述过程，client找到primary osd，并将Object写入到其上。
2. 再由primary osd写到 set中的secondary osd上。
3. 再由primary osd写到 set中的Tertiary osd上。2、3基本上是同时进行的。
4. Secondary osd接收到Object，写入后给Primary osd发送一个Ack，确认已经写入。
5. Tertiary osd接收到Object，写入后给Primary osd发送一个Ack，确认已经写入。4、5过程也可以是同时进行的。
6. Primary osd在接收到另两个都已经写入的确认后，向client发送一个Ack，表示此Object已经写入完毕。

至此一个Object的写过程就已经结束了。需要强调的是，上述1. ，2. ，3.都是发生在client上，而4.过程是client直接与Primary osd联系，减少了cluster中需要传输的总数据量。

### 使用纠删码

使用纠删码也只是上述过程中3、4步骤有所不同。



纠删码编码与解码的运算过程都是在Primary osd做的。一个Object通过Crush算法找到Primary osd，client完全写Object到Primary osd上。之后的过程如上图所示，根据纠删码的参数设置（K,M；K为数据块个数，M为编码块个数），Object在Primary osd上被切分成K个chunk或者说是fragment。若原来Object的大小不能被K整除，则会进行一些填充，以保证每个chunk大小相同。每个chunk的obj\_name都相同，用shardk来标记。

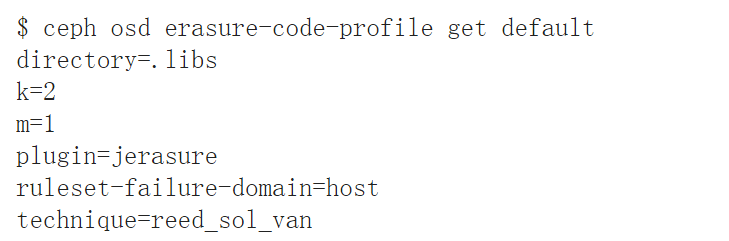
使用纠删码，原来的Object会产生K+M个chunk。第一个chunk存在Primary ods上，剩下的K+M-1个chunk，Primary osd会依次发送到相应的osd上。

在读文件时，Primary osd会收集相应osd上的chunk，然后在Primary osd恢复出原Object，发给client。Primary osd会优先收集数据块进行拼接重建出原Object，编码块用于恢复出丢失的块。

纠删码的具体配置过程可以参考：

<http://docs.ceph.com/docs/master/dev/erasure-coded-pool/>

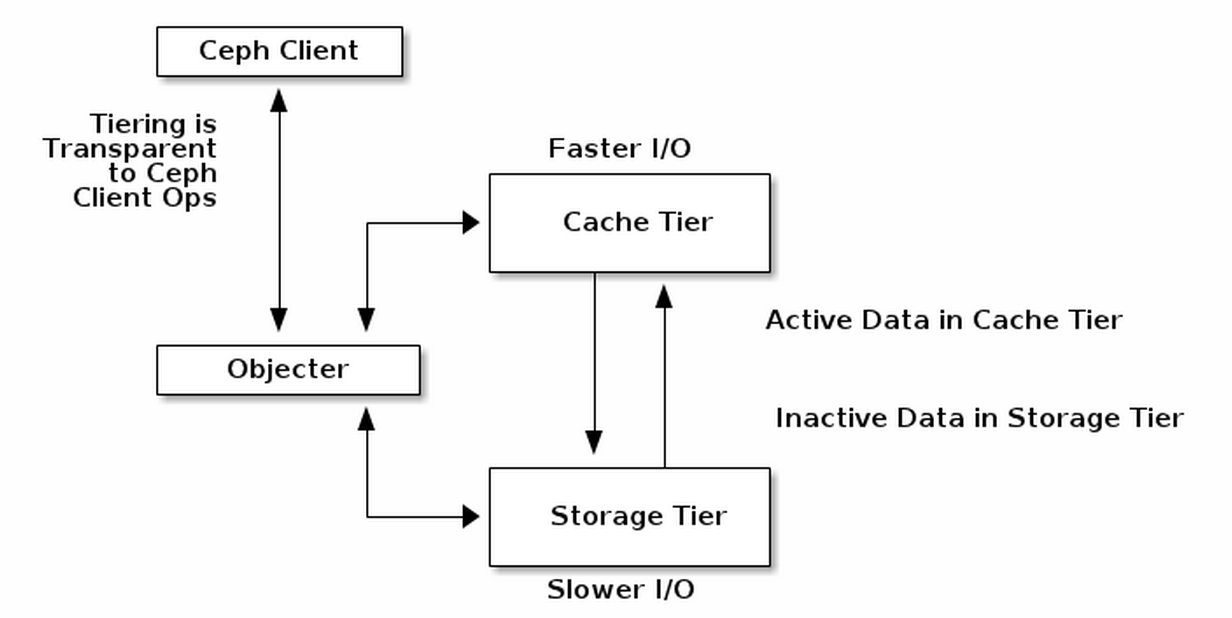
纠删码的默认参数为：



### 缓存分层

单独使用纠删码会产生较大程度的延迟，影响SLA。所以较新版本的ceph中都会将纠删码和缓存分层（Cache Tiering）结合起来使用。

缓存分层包括创建一个相对快速的存储设备池，比如说SSD，将其配置为一个缓存层。和一个后备的相对经济的存储设备层，该层主要放置经过纠删码编码后的数据块。Ceph objecter处理Objects的放置位置，tiering agent处理何时将Objects从cache写入到后备存储层。缓存层和后备存储层的存在相对于client都是透明的。



Cache tiering agent自动处理缓存层和后备存储层间的数据迁移。有两种模式可以选择：

* Writeback 模式：client写数据到缓存层，并从缓存层接收ACK，同时缓存层中的数据写入到存储层。当client需要存储层中的数据时，tiering agent会写先从存储层中读数据到缓存层，然后再发送给client。
* Read-only模式：client直接写数据到存储层。读数据时，tiering agent先将要读取的Objects从存储层复制到缓存层，再由缓存层发送给client。

要设置缓存分层就需要有两个池，一个作为后备存储，另一个作为cache。