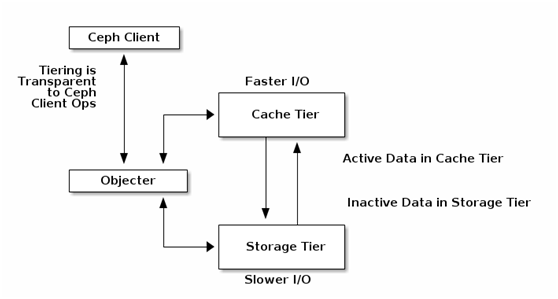
一、什么是cache tier

在ceph的手册里<http://docs.ceph.com/docs/master/rados/operations/cache-tiering/> 中对 cache tiering 进行了描述。

分布式的集群一般都是采用廉价的pc搭建，这些pc通常使用的是传统的机械硬盘，所以在磁盘的访问速度上有一定的限制，没有理想的iops数据。当去优化一个系统的IO性能时，最先想到的就是添加cache，热数据在cache被访问到，缩短数据的访问延时。Cache 一般会有 memory 或者ssd来做，考虑到价格和安全性，一般都是采用ssd作为cache。尤其实在随机io上，如果随机io全部放在ssd上，等数据冷却，将数据合并后，再刷入机械硬盘中，提升系统的io性能。(本来是想给大家看一下ssd和机械盘的性能测试结果的，但是测试数据是公司的不能放出来，自己测试的数据找不到了。。。。)。

对于ceph而言，怎么利用ssd作为普通磁盘的cache的，从手册中可知，先创建一个机械硬盘的pool，叫做storage tier，然后再创建一个使用ssd的pool ，叫做cache tier，把cache tier放在 storage tier之上。如下图：



当客户端访问操作数据时，优先读写cache tier数据(当然要根据cache mode来决定)，如果数据在storage tier 则会提升到cache tier中，在cache tier中 会有请求命中算法、缓存刷写算法、缓存淘汰算法等的实现，将热数据提升到cache tier中，将冷数据下放到storage tier中。

这就是一般的缓存技术实现，但是对于ceph来讲一个分布式的存储怎么来实现这个cache tier，实现起来是不是和文档一样的呢？让我们一起来探秘 cache tier.

二、  Cache tier 基本了解

1、cache tier 是基于pool的。这里值得注意的是cache pool 对应storage pool，不是ssd磁盘对应机械硬盘的，所以在cache tier和storage tier之间移动数据 是两个pool之间数据的移动，数据可能在不同地点的设备上移动。

2、cache mode有四种：writeback、forward、readonly、readforward、readproxy模式，这里每种模式都来解释下：

---a、writeback 模式:写操作，当请求到达cache成，完成写操作后，直接返回给客户端应答。后面由cache的agent线程负责将数据写入storage tier。读操作看是否命中缓存，如果命中直接在缓存读，没有命中可以redirect到storage tier访问。

---b、forward模式：所有的请求都redirct到storage tier 访问。

---c、readonly模式：写请求直接redirct到storage tier访问，读请求命中则直接处理，没有命中需要提升storage tier到cache tier中，完成请求，下次再读取直接命中缓存。

---d、readforward模式：读请求都redirect到storage tier中，写请求采用writeback模式。

---e、readproxy模式：读请求发送给cache tier，cache tier去base pool中读取，在cache tier获得object后，自己不保存，直接发送给客户端，写请求采用writeback模式。

3、cache的flush、evict算法，都是根据时间先后来的。这里有一些值得去改进的地方。

4、cache的agent 是应该根据pg的情况来实现，和pg的恢复流程一样，一个是recovery，一个是agent，但是最终放在了osd实现。

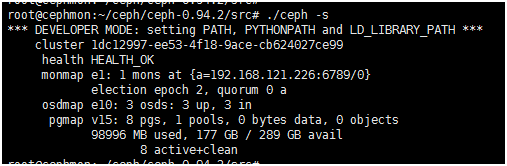
三、ceph cache tier处理流程：

还是原来的配方，正宗好凉cha。。。。说一大堆理论，也不如从代码入手，那我们就先从代码的流程来讲讲 cache tier，然后再分析cache tier的问题，最后觉得哪里可以完善的，希望和大家进行讨论。

手册里 已经教会大家怎么来使用这个cache tier的，不知道的自行回去修习，不然怕后面看不懂，那这里再手把手走一下？

我用的是实验环境，也没有ssd盘，直接在普通盘上创建一个cachepool，我们只是看一下cache tier的处理流程，所以影响不大。

1.1ceph 实验环境，可参考<https://my.oschina.net/u/2460844/blog/515353>。一个mon，3个osd。



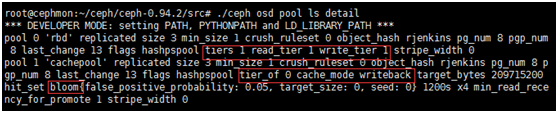
1.2创建cache pool，这里为了方便跟踪代码，所以创建的cachepool 只有8个pg。

这里原本有一个rbd pool是创建ceph默认，然后再新创建一个cache tier的 pool，新的pool名字直接叫做cachepool。现在我们有两个pool 一个是rbd，一个cachepool。

1.3 添加cachepool 为rbd的tier，这里直接使用add-cache 命令，该命令是其他几个命令集合的形式，用起来快速一点，采用默认项。但是后面要接一个缓存大小的设置，这里使用200M的大小。

./ceph osd tier add-cache rbd cachepool 209715200

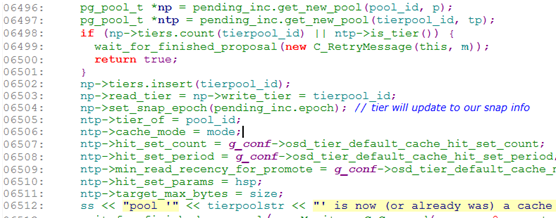
然后查看pool的详细信息



Pool 0 就是名字为rbd的pool，这里有两个重要指标 read\_tier 1 write\_tier 1,这里的1指的是pool 1. Pool 1就是刚刚创建的cachepool，在看看pool1 的设置，tier\_of 为0 是指pool1是pool0的tier，还有一个重要的要说cache\_mode，默认创建的就是write\_back模式的缓存，最后一个bloom 过滤器，缓存命中的算法，忘了是什么的人，可以搜一下。

代码里去搜一下add-cache的实现就可以知道了，命令会发送给osdminitor，在osd minitor中处理。

前面一些对pool的检查代码就不说了，这里是主要的代码，这部分代码比较简单可以一代而过就好了。总之就是将tier的关系添加到两个pool里，然后pool的信息会推送到所有的节点上。



6496：获取base pool信息

6497：获取cache pool 信息

6498：检查是pool tier的关系。

6502：将cache pool 添加到base pool 的tiers集合中。

6503：将base pool的read\_tier write\_tier都设置为cachepool。这个很有用。

6506：设置cache pool的缓存模式，默认为write\_back，可以手动设置其他模式。

6507~6511：就是和缓存 flush、evict相关的参数了。后面的讲flush和evict的时候会分析具体的参数。

到这里 已经设置好了 base pool和cache pool的tier关系了。那么就存在一个问题了，客户端要写一个object，这个object肯定是指向base pool的，怎么将object先写入到cache pool的？

1.4 测试读写请求，看请求如何从base pool转向到cache pool。

测试 代码，创建一个myimage的rbd，然后向rbd写入数据。

root@cephmon:~/ceph/ceph-0.94.2/src# cat ../python/create\_rbd.py

#!/usr/bin/env python

import sys,rados,rbd

def connectceph():

cluster = rados.Rados(conffile = '/root/ceph/ceph-0.94.2/src/ceph.conf')

cluster.connect()

ioctx = cluster.open\_ioctx('rbd')

rbd\_inst = rbd.RBD()

size = 1\*1024\*\*3 #4 GiB

rbd\_inst.create(ioctx,'myimage',size)

image = rbd.Image(ioctx,'myimage')

data = 'foo'\* 200

image.write(data,0)

image.close()

ioctx.close()

cluster.shutdown()

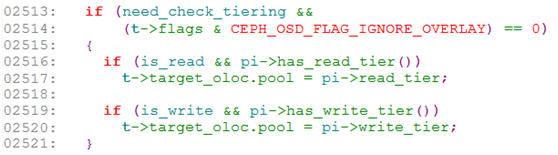
if \_\_name\_\_ == "\_\_main\_\_":

connectceph()

root@cephmon:~/ceph/ceph-0.94.2/src#

客户端的调用流程就不细说了，前面的blog也有讲过，这里我们挑重点的说。在客户端想ceph集群发送写请求的时候，一定会先拿到pool的相关信息，然后根据pool的信息再经过crush算法，得知数据副本所在的位置，然后把数据发送到对应的osd上。这里比较重要的就是计算crush算法，因为crush算法会决定你发送到那些osd上。

跟踪 客户端代码：…/src/osdc/Objecter.cc 中Objecter::\_calc\_target，捡重要的说，



2513：判断是否需要检查 存在cache tiering。

2516：如果是读请求，并且该pool存在read\_tier。这个read\_tier就是上面说的在osd monitor设置过的，指向cache pool。

2517：如果是读请求，则将target\_oloc.pool 设置为read\_tier。

2519：判断是写请求，并且该pool存在write\_tier。这个值也是在osd minitor中设置过的的，指向cache pool。

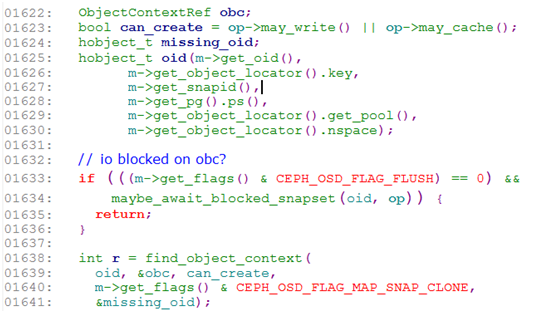
2518：如果是写请求，则将target\_oloc.pool 设置为write\_tier。

这里重要的是设置了target\_oloc.pool，本来这个值是指向base pool的，但是由于设置了add-cache 命令，这里target\_oloc.pool指向了cache pool。当相面使用target进行crush计算的时候，就能计算出cache pool中对应的object存储的osd。将请求发送到对应osd上。

1.5 cache pool的请求处理。决定请求是否缓存命中，或者redirect处理。

osd先接收到客户端发送来的请求，然后调用ReplicatedPG:: do\_request()  >>>>> ReplicatedPG::do\_op () 中处理，这都是正常的一个pool 处理请求的流程，在do\_op中来看看不同于其他普通pool的处理。

当然也在里面挑重要的说：

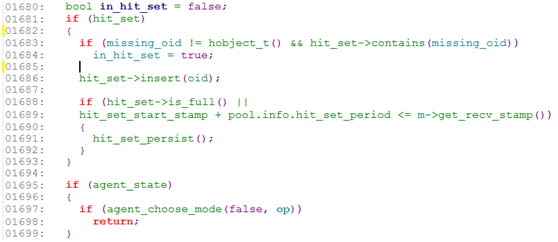


1623：判断下，当在对应的object 目录下找不到对应的object，是否需要创建object的context。如果是写请求需要创建，如果是读请求则不需要创建。

1625：根据接收到的msg恢复出 原始的oid（这个oid的信息仍然指向base pool，在1629中设置，但是当前处理的pg是cache pool的pg）。

1638：根据oid ，然后在cachepool中创建对应的head信息。Head和oid的区别在于一个指向了cache pool一个指向了base pool。查找head的context信息，如果是写操作没有找到head context，则会为head创建一个context，如果是读操作则没有则失败，有则返回正确。当读操作的时候，且查找context失败，则missing\_oid会指向head信息。

目前得到的几个信息整理下，oid(原始的object id)、missing\_oid(在cache pool中不存在的oid信息)，r(获取object context 失败错误码)，obc（object 的context信息）。继续下面的代码：



1680：当读操作时候，读的object 不再当前的cache pool中，有可能被evict 或者该object还没写入到cache中，但是该object曾经在操作过。

1681：判断是否有hit\_set 集合，该集合是用来统计pg中被操作过的object,无论是否命中过缓存。

1683：如果是读操作，且要读的object在当前的cachepool中不存在，但是在hit\_set中记录了该object 刚被访问过。

1684：如果刚刚同样也被访问过，则记录in\_hit\_set标记。（一定的时间内被访问多次）

1686：统计这次的访问操作object。

1688：如果这hit\_set 满了，或者时间间隔到了，则需要持久化这个hit\_set信息。

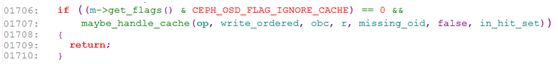
1691：hit\_set信息持久化。

1695：如果这个pg存在agent\_state

1697：这里判断缓存的flush、evict等操作，后面细说。Flush阈值、evict阈值的判断。只要不直接return，则不影响正常流程。

这里有两个重要的事情重申下，一个是hit\_set集合，一个是flush、evict等操作触发。

继续下面代码：

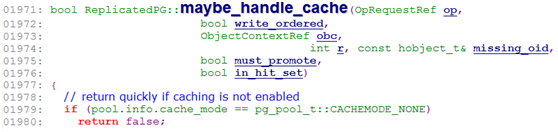


1706：判断msg中flags不需要 ignore cache的时候，则进行maybe\_handle\_cache()，如果maybe\_handle\_cache 处理成功了则直接return，否则继续进行后面的操作。

这个后面的操作就是基本的操作，如果读的话，则直接读本osd，如果写的话，就分发到其他replicate osd上。

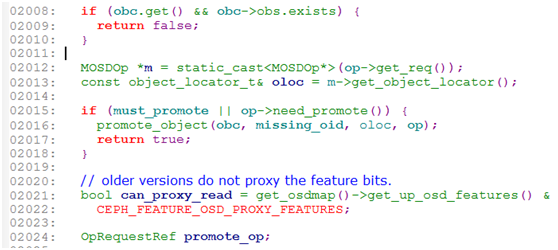
细看maybe\_handle\_cache做了什么，因为他影响了正常的处理流程。

ReplicatedPG::maybe\_handle\_cache()流程解析：



1971：函数入参解析，op就是当前需要处理的请求，write\_ordered 是否为写指令，obc 是object context，r是find\_object\_context 处理后的错误码，missing\_oid之前介绍过，must\_promote 是否要从base pool提升到cache pool，in\_hit\_set是否在近期刚访问过。

1979：判断该pool 如果不是cachepool 或者pool没有设置缓存模式，则认为不是缓存pool直接return false。普通pool或者cachepool未设置缓存模式，都直接返回。



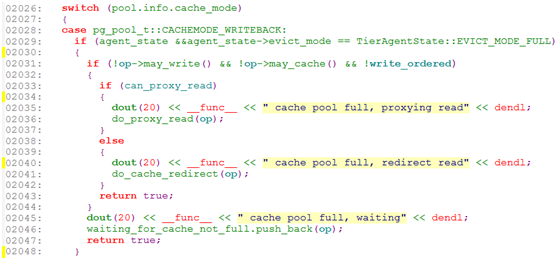
2008：好不起眼的一个判断，让人们仿佛忘了他的存在，但他确实坑爹一样的存在，为了表达它的重要性，你加个注释好么！ 这个判断就是为了判断该object是否在cache pool中是否命中。如果在cachepool中命中，则直接return false，然后在do\_op中会直接操作cache pool后面的流程。这里有个小问题，他没区分是cache模式 和op的操作因素，表示疑问，这部分后面也会有log进行证明。

2012：如果op没能命中cache pool则，则获取op中object的oloc信息，该信息记录了object所在的pool，现在它还指向base pool。

2015：如果设定了直接需要promote操作，则直接promote操作就可以了。这里再说一下promote\_object（）,是将base pool中对应的object拷贝到cache pool中，然后再处理op。这里只需要记住 promote\_object（）就是将object从base pool中拷贝到cache pool中即可。

2021：判断cachepool是否可以进行代理读 proxy\_read。默认是可以的，这个属性是后续版本加入的，ceph比较靠前的版本是没有的。

1.5.1 writeback模式：



2026：判断缓存模式

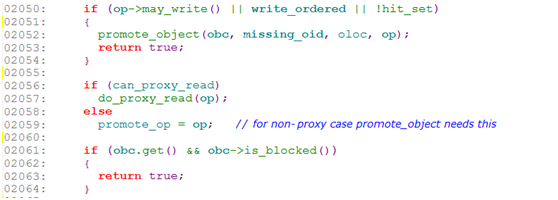
2028：如果是writeback模式，如果在未命中的情况下，writeback模式如何处理。

2029：如果现在cache已经满了，需要全部的evict。

2031：如果是读操作，可以proxy\_read的直接do\_proxy\_read(),否则do\_cache\_redirect()直接转向到base pool中。

2046：如果是写操作，添加到waiting\_for\_cache\_not\_full等待队列，缓存evict后会重新处理该op。

下面就是如果cache 在非full情况下的处理：

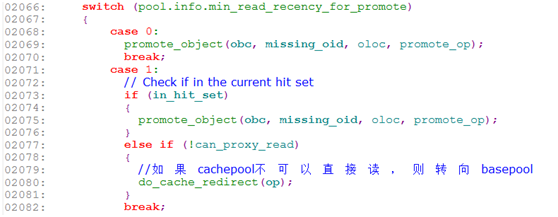


2050：如果是写请求的情况

2052：则需要从base pool  提升object到cache pool中。

2056：剩下后面的情况就都是读操作了，这里判断是否可以proxyread,如果可以则直接do\_proxy\_read()。

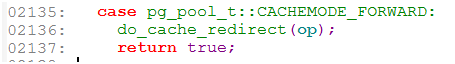
2061：如果这object是block住的，则不能进行操作。



在writeback模式下，读操作未命中时，min\_read\_recency\_for\_promote 该值默认情况下为1，这个在ceph osd pool ls detail 时也能看到。默认走了case1。如果近期访问过，说明object比较热，可以提升到cache 中，如果可以proxy read则在上面已经do\_proxy\_read(),如果不可以proxy\_read,则直接do\_cache\_redirect()。最后break ，return。

1.5.2  forward模式：

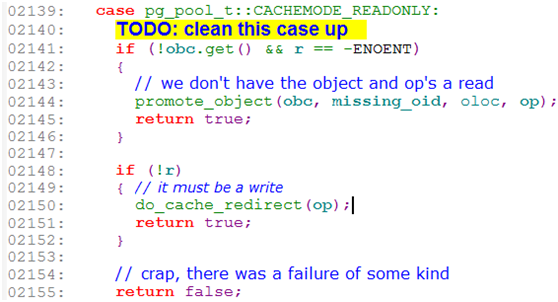
Forward模式就是把所有的请求都转向给base pool。



2135：如果是forward模式

2136：直接调用do\_cache\_redirect()直接告诉客户端，将请求转给base pool。

1.5.3  readonly 模式：



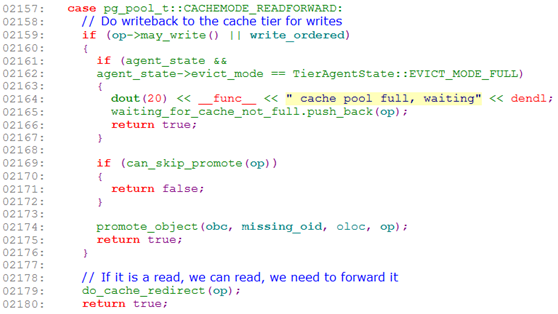
2139：case 当cache 的mode是readonly模式。

2141：如果不存在obc 并且没有创建，说明是读操作。在readonly模式时，读操作未命中的情况下需要promote\_object()。将object从base pool 提升到cache pool。

2148：如果未命中，但是find obc成功了，说明是未命中的写操作，所以r为0

2150：如果写操作未命中，在readonly模式下需要调用do\_cache\_redirect 将写操作转向到base pool。

1.5.4  readforward 模式：



2157：case 当cache 的mode是readforward模式。

2159：判断是否是写操作

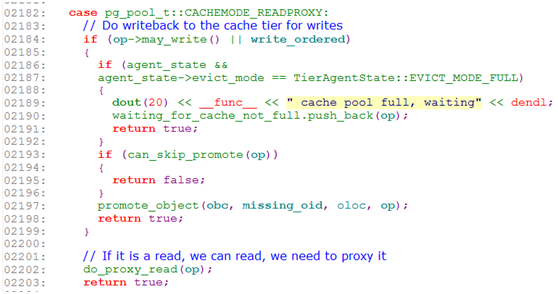
2161：如果pg存在agent\_state,并且evict\_mode是full模式，表示cache中的object都要evict。

2165：如果是evict full模式 ，就需要添加到等待队列中，等待evict完成，然后再处理。

2174：如果不是evict full 模式，则需要promote\_object()。

2179：如果是读操作，则直接转向到base pool。直接调用do\_cache\_redirect()

1.5.5  readproxy 模式：



2182：case 当cache 的mode是readproxy模式。

2184：判断是否是写操作

2186：如果pg存在agent\_state,并且evict\_mode是full模式，表示cache中的object都要evict。

2190：如果是evict full模式 ，就需要添加到等待队列中，等待evict完成，然后再处理。

2197：如果不是evict full 模式，则需要promote\_object()。

2202：如果是读操作，则使用代理读do\_proxy\_read()。

目前有几个地方还没有细说的，promote\_object()，do\_cache\_direct()，do\_proxy\_read()的操作。这几个代码比较简单，稍微熟悉就能看明白。

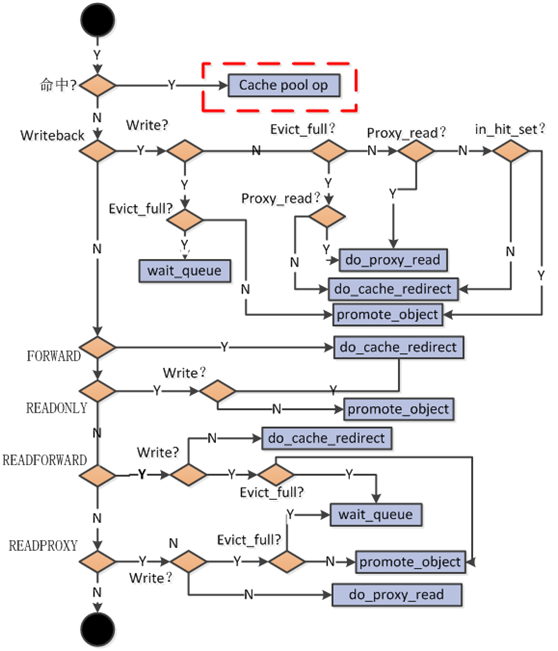
总结：

流程看着确实不复杂，能够捋顺关键点即可。

1、使用命令add-cache 可以将一个cachepool作为base pool的tier。这时会设置pool的信息，在pool里面记录了cache pool和base pool的关系。

2、客户端在获取pool信息的时候可知，目标base pool存在一个tier，叫做cache pool，那么操作base pool的请求都会发送给cache pool。

3、请求达到cache pool中时，作为tier的pool会有一些特别的处理maybe\_cache\_handle，具体的流程如下图：



a、会判断操作的object是否在cache pool中命中，如果命中，则直接在cache pool中处理，和在普通pool的请求一样处理。后续会有agent线程将缓存脏数据刷写到base pool中。

b、没有命中缓存的情况下，才会去判断缓存模式。如果命中缓存，不管是什么模式都会在cache pool中处理。下面的处理都是未命中缓存的情况。

c、判断是否是writeback模式，读操作，如果可以proxy\_read，那就直接do\_proxy\_read读取数据即可，不可以proxy\_read 就使用do\_cache\_redirect，告诉客户端去base pool中读取。写操作，如果当前是evict\_full模式，说明现在缓存中已经达到了阈值，需要等待缓存淘汰一些object，在完成写操作，目前放在等待队列中等待，如果不是evict\_full模式，则需要从base pool中promote对应的object到cache pool中，promote结束后继续处理本次的写操作。

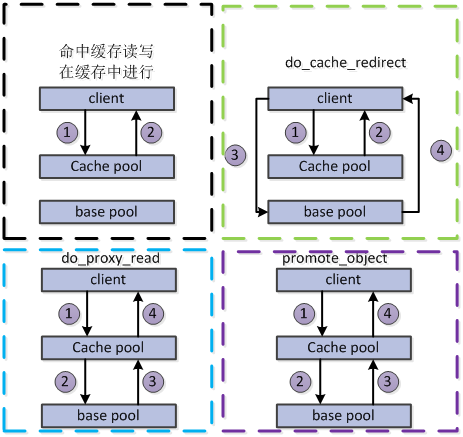
d、判断是否是forward模式。在forward模式下，不再在cachepool中处理请求，会告诉客户端将请求全部发送到base pool中。

e、判断是不是readonly模式。写操作会告诉客户端直接想base pool写即可，如果是读操作，则会从base pool中promote该object。

f、判断是不是readforward模式。该模式读操作全部都告诉客户端直接去base pool中读取即可，写操作按着writeback模式处理。

g、判断是不是readproxy模式。该模式读操作都采用cachepool的proxy read方法，写操作按着writeback模式处理。

h、这里在解释一下这几个操作如下图：



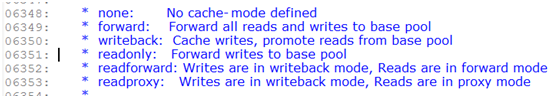
**绿框**：客户端请求cache pool，cache pool告诉客户端你应该去base pool中请求，客户端收到应答后，再次发送请求到base pool中请求数据，由base pool告诉客户端请求完成。

**蓝框**：客户端发送读请求到cache pool，但是未命中，则cache pool自己会发送请求到base pool中，获取数据后，由cache pool将数据发送给客户端，完成读请求。但是值得注意的是，虽然cache pool读取到了该object，但不会保存在cache pool中，下次请求仍然需要重新向basepool请求。

紫框：当客户端发送请求到cache pool中，但是cache pool未命中，cache pool会选择将该ojbect从base pool中提升到cache pool中，然后在cache pool进行读写操作，操作完成后告知客户端请求完成，在cache pool会缓存该object，下次直接在cache中处理，和proxy\_read存在的区别。

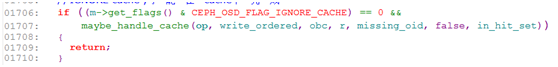
一个cache 重要不是他的数据处理流程，还有他的缓存flush策略、evict策略。下一节来细说一下cache的flush、evict如何来处理的。对本篇博客存在疑问的，欢迎批评指正。

对了，最后还有个疑问要说明下。就是你在writeback模式下，如果你的写操作命中缓存，则在cache pool中处理即可，但是readonly模式下，写操作也命中缓存，也会在cachepool中进行处理。而且在代码注释中写到OSDMinitor.cc中有这么一个注释



6351：readonly模式下，写操作会转向给base pool。但是在实际情况下抓取日志不是这样的。

日志的抓取，在ReplicatedPG::do\_op中找到maybe\_handle\_cache的调用处，



在1706之前添加一句，打印msg 的flags。表示要进入到了maybe\_handle\_cache().打印内容例如：”XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768”

在1710：之后添加一句“XYJdo\_opmaybe\_handle\_cache false”，如果日志中出现这句说明 命中缓存，直接在缓存中处理。

目的，要查看 writeback模式下，如果写操作命中缓存，会不会打印出上面的的输出“XYJdo\_opmaybe\_handle\_cache false”，如果在readonly模式下，不管写操作是否命中都应该转向给base pool，如果打印” XYJdo\_opmaybe\_handle\_cache false”，说明在readonly模式下缓存依然会处理写操作。

重新编译代码，启动实验环境，第一次设置为writeback模式：获取日志如下：

Line 402: 14:27:51.260842 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 403: 14:27:51.260860 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:0 ,r:-2 ,,missing\_oidfea29050/myimage.rbd/head//1 ,obc:0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 404: 14:27:51.260875 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache (no obc) missing\_oid fea29050/myimage.rbd/head//1 must\_promote 0 in\_hit\_set 0

Line 405: 14:27:51.260895 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache,write back:can\_proxy\_read,do\_proxy\_read

Line 418: 14:27:51.261328 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cachewrite\_back:min\_read\_recency\_for\_promote=1 in\_hit\_set:0 ,can\_proxy\_read1

Line 470: 14:27:51.270061 20 osd.1 pg\_epoch: 13 pg[1.4] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 471: 14:27:51.270068 10 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4f774b0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 472: 14:27:51.270090 25 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cache 30a98c1c/rbd\_directory/head//1(0'0 unknown.0.0:0 wrlock\_by=unknown.0.0:0 s 0 uv 0) DNE missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 498: 14:27:51.270616 20 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cachewrite back : may\_write:1 ,write\_ordered1 ,hit\_set:0x7fca6410baa0

Line 580: 14:27:51.273533 20 osd.1 pg\_epoch: 13 pg[1.4] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 582: 14:27:51.273541 10 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4f774b0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 583: 14:27:51.273550 25 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cache 30a98c1c/rbd\_directory/head//1(13'1 osd.1.0:1 wrlock\_by=unknown.0.0:0 whiteout s 0 uv 0) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 584: 14:27:51.273562 20 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 585: 14:27:51.273571 20 osd.1 pg\_epoch: 13 pg[1.4] XYJdo\_opmaybe\_handle\_cache false

Line 673: 14:27:51.330255 20 osd.1 pg\_epoch: 13 pg[1.4] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 674: 14:27:51.330263 10 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4f774b0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 675: 14:27:51.330273 25 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cache 30a98c1c/rbd\_directory/head//1(13'1 osd.1.0:1 wrlock\_by=unknown.0.0:0 whiteout s 0 uv 0) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 676: 14:27:51.330284 20 osd.1 pg\_epoch: 13 pg[1.4] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 677: 14:27:51.330292 20 osd.1 pg\_epoch: 13 pg[1.4] XYJdo\_opmaybe\_handle\_cache false

Line 835: 14:27:51.353307 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 836: 14:27:51.353314 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 837: 14:27:51.353323 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(0'0 unknown.0.0:0 wrlock\_by=unknown.0.0:0 s 0 uv 0) DNE missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 863: 14:27:51.353488 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cachewrite back : may\_write:1 ,write\_ordered1 ,hit\_set:0x7fca400d9ec0

Line 948: 14:27:51.356037 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 949: 14:27:51.356045 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 950: 14:27:51.356054 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(13'1 osd.1.0:3 wrlock\_by=unknown.0.0:0 whiteout s 0 uv 0) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 951: 14:27:51.356065 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 952: 14:27:51.356072 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1144: 14:27:51.418723 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1145: 14:27:51.418731 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1146: 14:27:51.418739 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(13'2 client.4120.0:4 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 679cf5ea) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1147: 14:27:51.418750 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 1148: 14:27:51.418756 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1182: 14:27:51.419938 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 1183: 14:27:51.419945 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1184: 14:27:51.419954 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(13'2 client.4120.0:4 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 679cf5ea) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1185: 14:27:51.419965 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 1186: 14:27:51.419972 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1327: 14:27:51.446803 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1328: 14:27:51.446810 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1329: 14:27:51.446819 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(13'3 client.4120.0:6 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 679cf5ea) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1330: 14:27:51.446829 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 1331: 14:27:51.446836 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1367: 14:27:51.447919 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1368: 14:27:51.447926 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1369: 14:27:51.447935 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(13'3 client.4120.0:6 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 679cf5ea) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1370: 14:27:51.447945 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 1371: 14:27:51.447952 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1412: 14:27:51.449095 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1413: 14:27:51.449102 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1414: 14:27:51.449110 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(13'3 client.4120.0:6 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 679cf5ea) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1415: 14:27:51.449120 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 1416: 14:27:51.449127 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1469: 14:27:51.450942 20 osd.1 pg\_epoch: 13 pg[1.7] XYJdo\_op maybe\_handle\_cache mesg->flags:4194341 IGNORE\_CACHE:32768

Line 1470: 14:27:51.450949 10 osd.1 pg\_epoch: 13 pg[1.7] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4e8a700 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1471: 14:27:51.450957 25 osd.1 pg\_epoch: 13 pg[1.7] maybe\_handle\_cache a9d57427/rb.0.1018.2ae8944a.000000000000/head//1(0'0 unknown.0.0:0 wrlock\_by=unknown.0.0:0 s 0 uv 0) DNE missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1497: 14:27:51.451280 20 osd.1 pg\_epoch: 13 pg[1.7] maybe\_handle\_cachewrite back : may\_write:1 ,write\_ordered1 ,hit\_set:0x7fca4004ac40

Line 1635: 14:27:51.453780 20 osd.1 pg\_epoch: 13 pg[1.7] XYJdo\_op maybe\_handle\_cache mesg->flags:4194341 IGNORE\_CACHE:32768

Line 1636: 14:27:51.453787 10 osd.1 pg\_epoch: 13 pg[1.7] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4e8a700 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1637: 14:27:51.453796 25 osd.1 pg\_epoch: 13 pg[1.7] maybe\_handle\_cache a9d57427/rb.0.1018.2ae8944a.000000000000/head//1(13'1 osd.1.0:6 wrlock\_by=unknown.0.0:0 whiteout s 0 uv 0) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1638: 14:27:51.453806 20 osd.1 pg\_epoch: 13 pg[1.7] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 1639: 14:27:51.453813 20 osd.1 pg\_epoch: 13 pg[1.7] XYJdo\_opmaybe\_handle\_cache false

Line 1838: 14:27:51.499268 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 1839: 14:27:51.499276 10 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x4fbb2d0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1840: 14:27:51.499284 25 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(13'3 client.4120.0:6 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 679cf5ea) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1841: 14:27:51.499294 20 osd.1 pg\_epoch: 13 pg[1.0] maybe\_handle\_cacheobc.get() && obc->obs.exists

Line 1842: 14:27:51.499301 20 osd.1 pg\_epoch: 13 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

清理环境，重新启动环境，第二次设置为readonly模式：获取日志如下：

Line 502: 14:42:02.383097 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 503: 14:42:02.383117 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:0 ,r:-2 ,,missing\_oidfea29050/myimage.rbd/head//1 ,obc:0 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 504: 14:42:02.383132 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache (no obc) missing\_oid fea29050/myimage.rbd/head//1 must\_promote 0 in\_hit\_set 0

Line 505: 14:42:02.383150 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cacheREADONLY

Line 506: 14:42:02.383156 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cacheREADONLY:!obc.get() && r == -ENOENT,promote\_object()

Line 632: 14:42:02.389038 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 633: 14:42:02.389047 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 634: 14:42:02.389056 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'1 osd.1.0:1 wrlock\_by=unknown.0.0:0 whiteout s 0 uv 0) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 635: 14:42:02.389068 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 636: 14:42:02.389078 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 731: 14:42:02.444435 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 732: 14:42:02.444443 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 733: 14:42:02.444452 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'1 osd.1.0:1 wrlock\_by=unknown.0.0:0 whiteout s 0 uv 0) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 734: 14:42:02.444463 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 735: 14:42:02.444470 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 837: 14:42:02.474868 20 osd.1 pg\_epoch: 15 pg[1.4] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 838: 14:42:02.474874 10 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3e30290 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 839: 14:42:02.474882 25 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cache 30a98c1c/rbd\_directory/head//1(0'0 unknown.0.0:0 wrlock\_by=unknown.0.0:0 s 0 uv 0) DNE missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 840: 14:42:02.474892 20 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cacheREADONLY

Line 841: 14:42:02.474899 20 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cacheREADONLY!r,r:0

Line 929: 14:42:02.494897 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 930: 14:42:02.494905 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 931: 14:42:02.494914 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'1 osd.1.0:1 wrlock\_by=unknown.0.0:0 whiteout s 0 uv 0) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 932: 14:42:02.494925 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 933: 14:42:02.494932 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1062: 14:42:02.534420 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1063: 14:42:02.534428 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1064: 14:42:02.534437 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'2 client.4121.0:5 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 35515d2f) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1065: 14:42:02.534448 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 1066: 14:42:02.534455 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1100: 14:42:02.535924 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 1101: 14:42:02.535932 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1102: 14:42:02.535940 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'2 client.4121.0:5 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 35515d2f) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1103: 14:42:02.535950 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 1104: 14:42:02.535957 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1245: 14:42:02.565499 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1246: 14:42:02.565507 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1247: 14:42:02.565515 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'3 client.4121.0:7 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 35515d2f) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1248: 14:42:02.565525 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 1249: 14:42:02.565532 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1285: 14:42:02.566794 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1286: 14:42:02.566802 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1287: 14:42:02.566813 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'3 client.4121.0:7 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 35515d2f) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1288: 14:42:02.566823 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 1289: 14:42:02.566830 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1330: 14:42:02.568110 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194321 IGNORE\_CACHE:32768

Line 1331: 14:42:02.568117 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:0 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1332: 14:42:02.568126 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'3 client.4121.0:7 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 35515d2f) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1333: 14:42:02.568136 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 1334: 14:42:02.568143 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

Line 1387: 14:42:02.570163 20 osd.1 pg\_epoch: 15 pg[1.4] XYJdo\_op maybe\_handle\_cache mesg->flags:4194341 IGNORE\_CACHE:32768

Line 1388: 14:42:02.570170 10 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3da8090 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1389: 14:42:02.570179 25 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cache 9105892c/rb.0.1019.238e1f29.000000000000/head//1(0'0 unknown.0.0:0 wrlock\_by=unknown.0.0:0 s 0 uv 0) DNE missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1390: 14:42:02.570190 20 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cacheREADONLY

Line 1391: 14:42:02.570196 20 osd.1 pg\_epoch: 15 pg[1.4] maybe\_handle\_cacheREADONLY!r,r:0

Line 1484: 14:42:02.587054 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_op maybe\_handle\_cache mesg->flags:4194340 IGNORE\_CACHE:32768

Line 1485: 14:42:02.587063 10 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache ,write:1 ,r:0 ,,missing\_oid0//0//-1 ,obc:0x3d72450 ,in\_hit\_set:0 ,pool.info.min\_read\_recency\_for\_promote:1

Line 1486: 14:42:02.587072 25 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache fea29050/myimage.rbd/head//1(15'3 client.4121.0:7 wrlock\_by=unknown.0.0:0 dirty|data\_digest s 112 uv 2 dd 35515d2f) exists missing\_oid 0//0//-1 must\_promote 0 in\_hit\_set 0

Line 1487: 14:42:02.587083 20 osd.1 pg\_epoch: 15 pg[1.0] maybe\_handle\_cache obc.get() && obc->obs.exists

Line 1488: 14:42:02.587090 20 osd.1 pg\_epoch: 15 pg[1.0] XYJdo\_opmaybe\_handle\_cache false

查看日志，不管是writeback模式还是readonly模式，在命中缓存时都会打印” maybe\_handle\_cache obc.get() && obc->obs.exists” 这句。从日志来看，在写操作的时候，如果命中缓存，后面都会接一句XYJdo\_opmaybe\_handle\_cache false，说明该写请求都会在缓存中处理。