CCeph PG状态

袁冬 2016年08月

# 1 PG与PG实例

PG表示一个逻辑概念，即数据分布相同的一组对象的集合，也可以理解为一致性哈希算法中的虚拟节点，即用作计算数据分布的单位。

一个存储池内的PG数目通常是恒定的，例如256个，每一个PG对应一个pg\_id。而每一个PG包含的对象数目可以是无限多个。对象通过一个简单的映射算法（例如余除）对应到PG。

一个PG在实际存储中，根据定义的副本数目，对应多个PG实例。例如，如果存储池采用三副本的存储策略，则每一个PG对应三个存储实例；如果存储池采用8+4的EC存储策略，则每一个PG对应12个存储实例。通常，将PG的实例数目记为N。

在计算PG的数据分布时，PG根据当前的OSD的情况，被映射到一个向量中，向量的维度为N。例如集群中存在osd.a, osd.b, …. Osd.z,对于pg.1，假设其N为3，即三副本方式存储，则pg.1的分布结果可能是(osd.d, osd.a, osd.m)，在这种情况下，pg.1的三个实例即pg.1a, pg.1b, pg.1c分别存储在osd.d, osd.a和osd.m上。

另外，在工程实现上，PG作为IO的处理单位，即每一个PG内部的IO是串行处理的，而不同PG之前的对象则是并行处理的。

# 2 PG实例状态

PG实例的主要状态包括：

* Initial：初始化状态
* Starting：处于从Initial到Active的过程中
* Active：活动状态，只有该状态下可以处理请求
* Disuse：无用状态，数据可能用于恢复，通常在恢复完成后会被删除
* Incomplete：数据不完整/数据丢失状态

基于PG的各个实例的状态，可以定义出PG的状态：

* Starting：PG的实例的状态是Initial、Starting、Active或Disuse；且处于Active状态PG实例数目不超过一半。
* Active：PG处于Active的实例数目超过一半。
* Incomplete：PG存在任一实例是InComplete状态。

定义：一个PG相对于某一个轮OSDMap是Active的，意味着在该轮OSDMap中，有>N/2个PG实例进入了Active状态。因为只有>N/2个PG实例进入了Active状态，该PG在该轮OSDMap中才有可能有请求成功。

# 3 PG状态的转换



Initial状态是PG的初始状态。在以下场景中，一个PG会被值为Initial状态：

* 当一个OSD启动的时候，其所有PG都被值为Initial状态
* 当一个PG在某个OSD上被创建时，被初始化为Initial状态
* 当OSDMap发生变化时，涉及到分布发生变化的所有PG都被置为Initial

Starting状态是一个中间状态，表示PG状态启动过程中；任意PG进入Initial后，都会进入Starting状态。

在Starting状态，PG会执行一系列操作，以便确定该PG最终能够进入到Active/Disuse/InComplete三个状态中的哪一个。

Active状态是PG的正常状态，即可以接受请求/对请求进行投票的状态。

一个PG实例进入了Active状态，表示：

1. 该PG获取了所有该PG的历史消息，并确认数据是完整的/可以获取到的
2. 该PG实例确认上一轮PG实例不可能再完成任何数据IO，即不会有两组处于不同OSDMap下的PG实例，同时位于Active状态（这样会导致数据异常）

Disuse状态是另一个PG的正常状态，即曾经是Active的PG实例，在当前OSDMap下不再是Active的。

但由于该PG实例上仍然存在数据恢复过程所需所需要的数据，因此尚不能删除时，此时PG实例的状态是Disuse。

作为Disuse状态的PG实例，其作用仅为了Active状态的PG实例的数据恢复提供数据，本身不再参与IO，类似一个旁观者的角色。

处于Disuse状态的PG实例，在Active状态的PG实例完成数据回复后，将被删除（由Active实例驱动）

InComplete状态表示PG实例无法获取到PG的完整数据，通常意味着该PG的数据已经暂时或永久性的丢失。

# 4 PG的启动流程

当处于Initial状态的PG实例收到EStart事件后，其状态转换为Starting，开始启动流程。

在启动过程中，PG实例首先根据OSDMap计算自身在本轮OSDMap（v）中是否是Active的，如果不是，则直接进入Disuse状态。如果PG实例在本轮OSDMap（v）中的角色是Active的，则PG实例尝试进入Active，具体流程为：

1. 与上一轮OSDMap（v-1)的PG实例进行通讯，即向所有上一轮OSDMap中该PG的所有实例发送GetPGInfo消息，目的是：
   1. 取得PG的信息包括：PGInfo和PGLog
   2. 通知上一轮PG实例停止接收请求
2. 假设在上一轮（v-1轮）OSDMap中，PG的实例数目为N，则当前PG实例必须收到N/2以上的PG实例的回复才可以进入Active状态。否则什么都不做（等待OSDMap的继续变化）
3. 当收到>N/2个回复后，如果在这个N/2个回复中，不存在任意Active的实例，则说明OSDMap（v-1)中整个PG没有进入Active状态，则可以忽略OSDMap(v-1)，继续向上回溯OSDMAP（v-2）.
4. 如果在收到的回复中，存在任意一个Active的PG实例，可以不再需要回溯OSDMap（v-2）。因为OSDMap（v-1）中存在一个Acitve的实例，则该实例必然已经和OSDMap（v-2）中的>N/2个实例进行通讯，即OSDMap（v-1）是有效的OSDMap
5. 当收到>N/2个回复，并且存在Active实例的回复时，继续启动流程：
   1. 计算PG的状态，包括PGInfo和PGLog
   2. 根据PG状态确认启动结果：
      1. 如果数据完整，则PG实例进入Active状态
      2. 如果数据不完整，则进入InComplete状态

当一个PG实例收到GetPGInfo消息后，其处理逻辑为：

1. 确认该消息对应的OSDMap版本是否比当前版本旧
   1. 如果是，忽略该消息；return
2. 放弃所有正在执行的请求/向客户端返回MOSDMapExpired
3. 将当前PG实例的状态值为Initial
4. 向消息来源返回PGInfo和PGLog信息

另外，如果PG实例收到了一个有效的GetPGInfo消息，即OSDMap版本比当前版本大，则说明当前OSD的OSDMap版本已经过旧，则需要通知OSD更新OSDMap（调用OSD的UpdateOSDMap方法）