CCeph PG状态

袁冬 2016年08月

# 1 PG状态

PG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，只用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

# 2 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，通常以为这该PG的数据已经暂时或永久性的丢失。

# 3 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PG的Active实例进行通讯（GetPGInfo消息）：
   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. 计算PG的状态，包括PGInfo和PGLog
3. 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版本已经过久，则需要通知OSD跟新OSDMap（调用OSD的UpdateOSDMap方法）