PG REMOVAL

See OSD::_remove_pg, OSD::RemoveWQ

There are two ways for a pg to be removed from an OSD:

1. MOSDPGRemove from the primary
2. OSD::advance_map finds that the pool has been removed

In either case, our general strategy for removing the pg is to atomically remove the metadata objects (pg->log_oid, pg->biginfo_oid) and rename the pg collections (temp, HEAD, and snap collections) into removal collections (see OSD::get_next_removal_coll). Those collections are then asynchronously removed. We do not do this inline because scanning the collections to remove the objects is an expensive operation. Atomically moving the directories out of the way allows us to proceed as if the pg is fully removed except that we cannot rewrite any of the objects contained in the removal directories until they have been fully removed. PGs partition the object space, so the only case we need to worry about is the same pg being recreated before we have finished removing the objects from the old one.

OSDService::deleting_pgs tracks all pgs in the process of being deleted. Each DeletingState object in deleting_pgs lives while at least one reference to it remains. Each item in RemoveWQ carries a reference to the DeletingState for the relevant pg such that deleting_pgs.lookup(pgid) will return a null ref only if there are no collections currently being deleted for that pg. DeletingState allows you to register a callback to be called when the deletion is finally complete. See PG::start_flush. We use this mechanism to prevent the pg from being “flushed” until any pending deletes are complete. Metadata operations are safe since we did remove the old metadata objects and we inherit the osr from the previous copy of the pg.

Similarly, OSD::osr_registry ensures that the OpSequencers for those pgs can be reused for a new pg if created before the old one is fully removed, ensuring that operations on the new pg are sequenced properly with respect to operations on the old one.

OSD::load_pgs() rebuilds deleting_pgs and osr_registry when scanning the collections as it finds old removal collections not yet removed.