## PUBLIC OSD VERSION

We maintain two versions on disk: an eversion_t pg_log.head and a version_t info.user_version. Each object is tagged with both the pg version and user_version it was last modified with. The PG version is modified by manipulating OpContext::at_version and then persisting it to the pg log as transactions, and is incremented in all the places it used to be. The user_version is modified by manipulating the new OpContext::user_at_version and is also persisted via the pg log transactions. user_at_version is modified only in PrimaryLogPG::prepare_transaction when the op was a “user modify” (a non-watch write), and the durable user_version is updated according to the following rules: 1) set user_at_version to the maximum of ctx->new_obs.oi.user_version+1 and info.last_user_version+1. 2) set user_at_version to the maximum of itself and ctx->at_version.version. 3) ctx->new_obs.oi.user_version = ctx->user_at_version (to change the object’s user_version)

This set of update semantics mean that for traditional pools the user_version will be equal to the past reassert_version, while for caching pools the object and PG user-version will be able to cross pools without making a total mess of things. In order to support old clients, we keep the old reassert_version but rename it to “bad_replay_version”; we fill it in as before: for writes it is set to the at_version (and is the proper replay version); for watches it is set to our user version; for ENOENT replies it is set to the replay version’s epoch but the user_version’s version. We also now fill in the version_t portion of the bad_replay_version on read ops as well as write ops, which should be fine for all old clients.

For new clients, we prevent them from reading bad_replay_version and add two proper members: user_version and replay_version; user_version is filled in on every operation (reads included) while replay_version is filled in for writes.

The objclass function get_current_version() now always returns the pg->info.last_user_version, which means it is guaranteed to contain the version of the last user update in the PG (including on reads!).