BookieRecoveryPage

Bookie Recovery Design

Problem statement and trade-offs

The essential idea of the bookie recovery feature is to enable an application to heal its bookie ensemble once some bookie has crashed. The bookie recovery task is basically the one of reconstructing the ledger fragment that the crashed bookie stored or should have stored, had it not crashed.

By design, a bookie can store fragments of multiple ledgers. To recover a bookie, we hence need to create new copies of each of the fragments that were present in the faulty bookie. There are two choices: we recover ledgers individually or we recover one ledger at a time. To decide which one is more appropriate, we have to think about how we will use such a recovery tool. If applications are to run such a tool, then it is probably best to recover one at a time or using small batches. If some operations team performs recovery on behalf of applications, then they will probably prefer to recover the whole set of faulty bookie.

Such a recovery tool can run either as a separate client or directly in a bookie. The advantage of implementing recovery on the client side is simplicity: we can just leverage the client implementation to read entries and write to the new bookie. Performing such a task in a client, however, may lead to an inefficient utilization of network bandwidth. For an efficient utilization of network bandwidth, it is best to copy entries directly.

After executing such a recovery procedure, one expects to have valid copies of the entries in the new bookie(s). To validate entries, a user needs the secret that was used to write to the ledger. It becomes a concern then to use such secrets if one is to perform recovery of ledgers in batches, as such a user needs to know all secrets of all ledgers to be recovered. We can, however, separate concerns and make copying separate from validation. That is, we could have two separate tools, one for copying and another for data integrity validation.

Design choices (version 0.1)

Our current plan is to use a client to perform recovery of a bookie. The high-level idea is that a client calls recover(bookie id), and executes the recovery procedure. In slightly more detail, here is the pseudo-code:

```
recover(bid)
For each ledger lid of bid
   Select new bookie nbid
   Use bk client to read entries of lid that correspond to bid
   Write to *nbid*
```