

BP-1 - 64 bits ledger id support

Status

Current state: *Done*

Released: 4.5.0

Problem

- the interface, wire protocol and storage all support 64 bits ledger id.
- however, the ledger id generation on zookeeper only generate 32 bits ledger id.
- the zookeeper-based ledger metadata store uses 2-4-4 split to store 32 bits ledger id.
- we need to find a solution to support 64 bits in zookeeper-based ledger metadata store.

Proposal

- there are 3 major concerns for this problem
 - how to store metadata for 64-bits-id ledgers in zookeeper?
 - how to generate 64-bits ledger id using zookeeper?
 - how to achieve backward compatibility?
- how to store?
 - Salesforce built a Long Hierarchical Ledger Manager
 - Ledger ID is split in 3-4-4-4-4 components.
 - No backward compatible on initial built
 - backward compatibility?
 - for ledger id whose higher 32 bits are zero, use split 2-4-4
 - for ledger id whose higher 32 bits are higher than zero, use split 3-4-4-4-4
 - since the first component has different characters (2 vs 3), it is easy to achieve backward compatibility
 - negative ledger id discussion
- how to generate?
 - use zookeeper sequence znode to create lower 32 bits
 - use another znode to create sequential buckets, the buckets are used for higher 32 bits.
 - when to create bucket?
 - when lower 32-bits generation is approaching 31 bits, create a new bucket.
 - creating a new bucket and the id generation will happen under one zookeeper 'multi' transaction call to guarantee there is only one bucket is created.

Action

- [BOOKKEEPER-552: 64 Bits Ledger ID Generation](#)
- [BOOKKEEPER-553: New LedgerManager for 64 Bits Ledger ID Management in ZooKeeper](#)