Item 322

git_comments:

1. * * * Ledger-Op (Create/Delete) timeout * * @return
2. * * Ledger-Op (Create/Delete) timeout after which callback will be completed with failure * * @param metadataOperationsTimeoutSeconds
3. * * check if ledger-op task is already completed by timeout-task. If completed then delete the created ledger * * @param rc * @param lh * @param ctx * @return
4. * * Create ledger async and schedule a timeout task to check ledger-creation is complete else it fails the callback * with TimeoutException. * * @param bookKeeper * @param config * @param digestType * @param cb * @param emptyMap
5. ledger-creation is already timed out and callback is already completed so, delete this ledger and return.
6. operation timeout while updating managed-ledger metadata.

git_commits:

1. **summary:** Add ledger op timeout to avoid topics stuck on ledger-creation (#2535)
   message: Add ledger op timeout to avoid topics stuck on ledger-creation (#2535) * Add ledger op timeout to avoid topics stuck on ledger-creation * rename to metadataOperationsTimeoutSeconds * ad service config for managedLedgerMetadataOperationsTimeoutSeconds

github_issues:

github_issues_comments:

github_pulls:

1. **title:** Add ledger op timeout to avoid topics stuck on ledger-creation
   body: ### Motivation Frequently, we are having issue where many topics across the all brokers get stuck on creating-ledger state and create-ledger-callback never gets completed. Right now, it mainly happens when 1. Anytime ZK leader restarts 2. ZK quorum restarts due to [32 bit rollover](https://jira.apache.org/jira/browse/ZOOKEEPER-1278) In such cases, zk-client doesn't complete the callback and broker keeps waiting with creating-ledger state until we unload the topic. It happens frequently to us and it's hard to catch such stuck topics and it requires roll-over broker restart. So, Broker should have a way to timeout LedgerOp (create/delete) and complete callback with timeout-exception. ### Modifications Add Ledger-Op timeout configuration, and managed-ledger fails callback if ledger-creation doesn't complete in that time duration. ### Result Broker can recover stuck managed-ledger which are waiting on ledger creation callback.
2. **title:** Add ledger op timeout to avoid topics stuck on ledger-creation
   body: ### Motivation Frequently, we are having issue where many topics across the all brokers get stuck on creating-ledger state and create-ledger-callback never gets completed. Right now, it mainly happens when 1. Anytime ZK leader restarts 2. ZK quorum restarts due to [32 bit rollover](https://jira.apache.org/jira/browse/ZOOKEEPER-1278) In such cases, zk-client doesn't complete the callback and broker keeps waiting with creating-ledger state until we unload the topic. It happens frequently to us and it's hard to catch such stuck topics and it requires roll-over broker restart. So, Broker should have a way to timeout LedgerOp (create/delete) and complete callback with timeout-exception. ### Modifications Add Ledger-Op timeout configuration, and managed-ledger fails callback if ledger-creation doesn't complete in that time duration. ### Result Broker can recover stuck managed-ledger which are waiting on ledger creation callback.
3. **title:** Add ledger op timeout to avoid topics stuck on ledger-creation
   body: ### Motivation Frequently, we are having issue where many topics across the all brokers get stuck on creating-ledger state and create-ledger-callback never gets completed. Right now, it mainly happens when 1. Anytime ZK leader restarts 2. ZK quorum restarts due to [32 bit rollover](https://jira.apache.org/jira/browse/ZOOKEEPER-1278) In such cases, zk-client doesn't complete the callback and broker keeps waiting with creating-ledger state until we unload the topic. It happens frequently to us and it's hard to catch such stuck topics and it requires roll-over broker restart. So, Broker should have a way to timeout LedgerOp (create/delete) and complete callback with timeout-exception. ### Modifications Add Ledger-Op timeout configuration, and managed-ledger fails callback if ledger-

creation doesn't complete in that time duration. ### Result Broker can recover stuck managed-ledger which are waiting on ledger creation callback.

4. **title:** Add ledger op timeout to avoid topics stuck on ledger-creation
   body: ### Motivation Frequently, we are having issue where many topics across the all brokers get stuck on creating-ledger state and create-ledger-callback never gets completed. Right now, it mainly happens when 1. Anytime ZK leader restarts 2. ZK quorum restarts due to [32 bit rollover] (https://jira.apache.org/jira/browse/ZOOKEEPER-1278) In such cases, zk-client doesn't complete the callback and broker keeps waiting with creating-ledger state until we unload the topic. It happens frequently to us and it's hard to catch such stuck topics and it requires roll-over broker restart. So, Broker should have a way to timeout LedgerOp (create/delete) and complete callback with timeout-exception. ### Modifications Add Ledger-Op timeout configuration, and managed-ledger fails callback if ledger-creation doesn't complete in that time duration. ### Result Broker can recover stuck managed-ledger which are waiting on ledger creation callback.
5. **title:** Add ledger op timeout to avoid topics stuck on ledger-creation
   body: ### Motivation Frequently, we are having issue where many topics across the all brokers get stuck on creating-ledger state and create-ledger-callback never gets completed. Right now, it mainly happens when 1. Anytime ZK leader restarts 2. ZK quorum restarts due to [32 bit rollover] (https://jira.apache.org/jira/browse/ZOOKEEPER-1278) In such cases, zk-client doesn't complete the callback and broker keeps waiting with creating-ledger state until we unload the topic. It happens frequently to us and it's hard to catch such stuck topics and it requires roll-over broker restart. So, Broker should have a way to timeout LedgerOp (create/delete) and complete callback with timeout-exception. ### Modifications Add Ledger-Op timeout configuration, and managed-ledger fails callback if ledger-creation doesn't complete in that time duration. ### Result Broker can recover stuck managed-ledger which are waiting on ledger creation callback.
   label: code-design
6. **title:** Add ledger op timeout to avoid topics stuck on ledger-creation
   body: ### Motivation Frequently, we are having issue where many topics across the all brokers get stuck on creating-ledger state and create-ledger-callback never gets completed. Right now, it mainly happens when 1. Anytime ZK leader restarts 2. ZK quorum restarts due to [32 bit rollover] (https://jira.apache.org/jira/browse/ZOOKEEPER-1278) In such cases, zk-client doesn't complete the callback and broker keeps waiting with creating-ledger state until we unload the topic. It happens frequently to us and it's hard to catch such stuck topics and it requires roll-over broker restart. So, Broker should have a way to timeout LedgerOp (create/delete) and complete callback with timeout-exception. ### Modifications Add Ledger-Op timeout configuration, and managed-ledger fails callback if ledger-creation doesn't complete in that time duration. ### Result Broker can recover stuck managed-ledger which are waiting on ledger creation callback.
7. **title:** Add ledger op timeout to avoid topics stuck on ledger-creation
   body: ### Motivation Frequently, we are having issue where many topics across the all brokers get stuck on creating-ledger state and create-ledger-callback never gets completed. Right now, it mainly happens when 1. Anytime ZK leader restarts 2. ZK quorum restarts due to [32 bit rollover] (https://jira.apache.org/jira/browse/ZOOKEEPER-1278) In such cases, zk-client doesn't complete the callback and broker keeps waiting with creating-ledger state until we unload the topic. It happens frequently to us and it's hard to catch such stuck topics and it requires roll-over broker restart. So, Broker should have a way to timeout LedgerOp (create/delete) and complete callback with timeout-exception. ### Modifications Add Ledger-Op timeout configuration, and managed-ledger fails callback if ledger-creation doesn't complete in that time duration. ### Result Broker can recover stuck managed-ledger which are waiting on ledger creation callback.

github_pulls_comments:

1. @merlimat can you please review it as we want to use this patch in our current deployed-release.
2. @merlimat fixed it.
3. rerun integration tests

github_pulls_reviews:

1. **body:** Maybe use a more explicit name like `metadataOperationsTimeoutSeconds` ? Also, the default in `ServiceConfiguration` seems to be 60sec
   label: code-design

2. sure, let me rename it. > Also, the default in ServiceConfiguration seems to be 60sec Sure, I can make it 60 sec but in this PR, default time we are setting as `ZooKeeperSessionTimeout (default 30sec) *2` as it mainly happens when zk-client fails to complete callback when zk leader restarts.

jira_issues:

1. **summary:** acceptedEpoch not handling zxid rollover in lower 32bits
   description: When the lower 32bits of a zxid "roll over" (zxid is a 64 bit number, however the upper 32 are considered the epoch number) the epoch number (upper 32 bits) are incremented and the lower 32 start at 0 again. This should work fine, however, afaict, in the current 3.4/3.5 the acceptedEpoch/currentEpoch files are not being updated for this case. See ZOOKEEPER-335 for changes from 3.3 branch.
2. **summary:** acceptedEpoch not handling zxid rollover in lower 32bits
   description: When the lower 32bits of a zxid "roll over" (zxid is a 64 bit number, however the upper 32 are considered the epoch number) the epoch number (upper 32 bits) are incremented and the lower 32 start at 0 again. This should work fine, however, afaict, in the current 3.4/3.5 the acceptedEpoch/currentEpoch files are not being updated for this case. See ZOOKEEPER-335 for changes from 3.3 branch.
3. **summary:** acceptedEpoch not handling zxid rollover in lower 32bits
   description: When the lower 32bits of a zxid "roll over" (zxid is a 64 bit number, however the upper 32 are considered the epoch number) the epoch number (upper 32 bits) are incremented and the lower 32 start at 0 again. This should work fine, however, afaict, in the current 3.4/3.5 the acceptedEpoch/currentEpoch files are not being updated for this case. See ZOOKEEPER-335 for changes from 3.3 branch.
4. **summary:** acceptedEpoch not handling zxid rollover in lower 32bits
   description: When the lower 32bits of a zxid "roll over" (zxid is a 64 bit number, however the upper 32 are considered the epoch number) the epoch number (upper 32 bits) are incremented and the lower 32 start at 0 again. This should work fine, however, afaict, in the current 3.4/3.5 the acceptedEpoch/currentEpoch files are not being updated for this case. See ZOOKEEPER-335 for changes from 3.3 branch.
5. **summary:** acceptedEpoch not handling zxid rollover in lower 32bits
   description: When the lower 32bits of a zxid "roll over" (zxid is a 64 bit number, however the upper 32 are considered the epoch number) the epoch number (upper 32 bits) are incremented and the lower 32 start at 0 again. This should work fine, however, afaict, in the current 3.4/3.5 the acceptedEpoch/currentEpoch files are not being updated for this case. See ZOOKEEPER-335 for changes from 3.3 branch.

jira_issues_comments:

1. I just tested this with my test from ZOOKEEPER-1277 and it fails with out the hzxid change in ZooKeeperServer. However even with that patch it still fails, I'm assuming because the acceptedEpoch, etc... files are not being updated properly. Camille can you take a look?
2. This patch passes the simple test, however the others all fail. This is the test/fix from ZOOKEEPER-1277
3. This turns out to be a duplicate of ZOOKEEPER-1277 - that patch causes the leader to be re-elected just prior to rollover. 1277 was applied to 3.3/3.4/3.5(trunk)