

git_comments:

1. **** Gets and clears queuedType**
2. **** Mark pending operation PARTIAL if not full and return true if there is no in-progress operation ** @return True if we should execute a scan, False if there is already one running**
3. **** Performs scan described by instanceState using the cached bookie information * in bookieInfoMap.**
4. **** Removes bookie from bookieInfoMap ** @param bookie bookie on which we observed an error**
5. **** Mark pending operation FULL and return true if there is no in-progress operation ** @return True if we should execute a scan, False if there is already one running**
6. **** Tracks whether there is an execution in progress as well as whether * another is pending.**
7. **We reread bookies missing from the map each time, so remove to ensure we get to it on the next scan**
8. **** Method to allow tests to block until bookie info is available ** @param bookie to lookup * @return None if absent, free disk space if present**
synchronized Optional<Long> getFreeDiskSpace(BookieSocketAddress bookie) { BookieInfo bookieInfo = bookieInfoMap.getInfo(bookie); if (bookieInfo != null) { return Optional.of(bookieInfo.getFreeDiskSpace()); return Optional.empty(); } } /* State to track scan execution progress as callbacks come in
9. **** Returns info for bookie, null if not known ** @param bookie bookie for which to get info * @return Info for bookie, null otherwise**
10. **** Tracks the most recently reported set of bookies from BookieWatcher as well * as current BookieInfo for bookies we've successfully queried.**
11. **** Contains the most recently reported set of bookies from BookieWatcher * A partial query consists of every member of mostRecentlyReportedBookies * minus the entries in bookieInfoMap.**
12. **** Contains the most recently obtained information on the contained bookies. * When an error happens querying a bookie, the entry is removed.**
13. **** Get bookie info map**
14. **** If queuedType != UNQUEUED, returns true, leaves running equal to true * Otherwise, returns false and sets running to false**
15. **** Report new info on bookie ** @param bookie bookie for which we obtained new info * @param info the new info**
16. **** Set the time interval between retries on unsuccessful GetInfo requests * * * @param interval * @param unit * @return client configuration**
17. **** Get the time interval between retries on unsuccessful bookie info request. Default is * 60s. ** @return**
18. **after delaySecs, advertise finalFreeDiskSpace; before that advertise initialFreeDiskSpace**
19. **Sleep for double the time required to update the bookie infos, and then check each one**
20. **the start of the test, and 3MB once useHigherValue is set**
21. **** Kill bookie by index and verify that it's stopped ** @param index index of bookie to kill ** @return configuration of killed bookie**

git_commits:

1. **summary:** BOOKKEEPER-1102: Clarify BookieInfoReader and fix associated test flappers
message: BOOKKEEPER-1102: Clarify BookieInfoReader and fix associated test flappers BookieInfoReader: The previous synchronization logic wasn't really correct, and the logic at the top of the method was far more complicated than it needed to be. Restrict bookies to be non-null. Restructure the code to simply use the BookieInfoReader instance as a single lock. One significant behavioral change is that we scan every bookie not in the map, and we clear from the map bookies which returned an error. Also, explicitly cache the most recent bookie set reported by the BookieWatcher. This eliminates the need to call into BookieWatcher from getReadWriteBookieInfo and the corresponding error path. The periodic scan continues to explicitly check. Another departure is the addition of an explicit retry-on-error param to trigger retry if any of the requests failed (getBookieInfoRetryIntervalSeconds). We'll only retry the ones that actually failed (along with any new additions since the last run). This is useful because bookie startup triggers the addition of the bookie node to zk before the bookie actually becomes available for the bookie info request, so there can be rare races in the unit tests between BookieInfoReader requesting the info and the bookie actually being up. Also, add a method to allow tests to wait for updates to be reflected. PerChannelBookieClient: fix error handling for BookieInfo keys Passing a key corresponding to a GET_BOOKIE_INFO operation to errorOutReadKey results in a casting exception, clean up the invalid calls. BookKeeperClusterTestCase: add killBookieAndWaitForZK Should reduce the need for tests to wait for an arbitrary period to let the cluster "settle". BookKeeperDiskSpaceWeightedLedgerPlacementTest: This test was heavily time dependent, and the Thread.sleep values did not work universally. Instead, eliminate the arbitrary Thread.sleep values and instead verify the free space changes on each change. Also, switch the delay on testDiskSpaceWeightedBookieSelectionWithPeriodicBookieInfoUpdate to simply use an atomic boolean to signal the value switch. Signed-off-by: Samuel Just <sjust@salesforce.com> Author: Samuel Just <sjust@salesforce.com> Reviewers: Enrico Olivelli <eolivelli@gmail.com> This patch had conflicts when merged, resolved by Committer: Sijie Guo <sijie@apache.org> This closes #275 from athanatos/forupstream/BOOKKEEPER-1102
label: code-design

github_issues:**github_issues_comments:****github_pulls:**

1. **title:** BOOKKEEPER-1102: Clarify BookieInfoReader and fix associated test flappers
body: BookieInfoReader: The previous synchronization logic wasn't really correct, and the logic at the top of the method was far more complicated than it needed to be. Restrict bookies to be non-null. Restructure the code to simply use the BookieInfoReader instance as a single lock. One significant behavioral change is that we scan every bookie not in the map, and we clear from the map bookies which returned an error. Also, explicitly cache the most recent bookie set reported by the BookieWatcher. This eliminates the need to call into BookieWatcher from getReadWriteBookieInfo and the corresponding error path. The periodic scan continues to explicitly check. Another departure is the addition of an explicit retry-on-error param to trigger retry if any of the requests failed (getBookieInfoRetryIntervalSeconds). We'll only retry the ones that actually failed (along with any new additions since the last run). This is useful because bookie startup triggers the addition of the bookie node to zk before the bookie actually becomes available for the bookie info request, so there can be rare races in the unit tests between BookieInfoReader requesting the info and the bookie actually being up. Also, add a method to allow tests to wait for updates to be reflected. PerChannelBookieClient: fix error handling for BookieInfo keys Passing a key corresponding to a GET_BOOKIE_INFO operation to errorOutReadKey results in a casting exception, clean up the invalid calls. BookKeeperClusterTestCase: add killBookieAndWaitForZK Should reduce the need for tests to wait for an arbitrary period to let the cluster "settle". BookKeeperDiskSpaceWeightedLedgerPlacementTest: This test was heavily time dependent, and the Thread.sleep values did not work universally. Instead, eliminate the arbitrary Thread.sleep values and instead verify the free space changes on each change. Also, switch the delay on testDiskSpaceWeightedBookieSelectionWithPeriodicBookieInfoUpdate to simply use an atomic boolean to signal the value switch. Signed-off-by: Samuel Just <sjust@salesforce.com>
label: code-design

github_pulls_comments:

1. **body:** I think that this patch does more than fixing a test case. Maybe we should change the title
label: test

2. @athanatos is this your current code at Salesforce or is it just a proposal ? as far as I know you are the only current user of DiskSpaceWeightedLedgerPlacement functions
3. @eolivelli It is our current code at SF as of last week.
4. @athanatos could you address @sijie comments ? the patch is almost ready to go
5. @eolivelli Will do today.
6. @sijie @eolivelli Ok, updated.
7. Seems there is a conflicting file.

github_pulls_reviews:

1. is there any reason not using an Enum?
2. **body:** any reason the put exception as the first parameter? it is a common practice in slf4j to put exception as the last parameter, and you don't need '{}' for exceptions.
label: code-design

jira_issues:

1. **summary:**
org.apache.bookkeeper.client.BookKeeperDiskSpaceWeightedLedgerPlacementTest.testDiskSpaceWeightedBookieSelectionWithBookiesBeingAdded is unreliable
description:
org.apache.bookkeeper.client.BookKeeperDiskSpaceWeightedLedgerPlacementTest.testDiskSpaceWeightedBookieSelectionWithBookiesBeingAdded can intermittently fail depending on the timing of the client receiving the info back from the bookies. Additionally, the synchronization in BookieInfoReader is more complicated than necessary and not entirely correct.

jira_issues_comments:

1. Issue resolved by merging pull request 275 [<https://github.com/apache/bookkeeper/pull/275>] {noformat} commit a5f8580f53464065243a9af038935f5893434166 Author: Samuel Just <sjust@salesforce.com> AuthorDate: Mon Jul 31 23:34:07 2017 -0700 Commit: Sijie Guo <sijie@apache.org> CommitDate: Mon Jul 31 23:34:07 2017 -0700 BOOKKEEPER-1102: Clarify BookieInfoReader and fix associated test flappers BookieInfoReader: The previous synchronization logic wasn't really correct, and the logic at the top of the method was far more complicated than it needed to be. Restrict bookies to be non-null. Restructure the code to simply use the BookieInfoReader instance as a single lock. One significant behavioral change is that we scan every bookie not in the map, and we clear from the map bookies which returned an error. Also, explicitly cache the most recent bookie set reported by the BookieWatcher. This eliminates the need to call into BookieWatcher from getReadWriteBookieInfo and the corresponding error path. The periodic scan continues to explicitly check. Another departure is the addition of an explicit retry-on-error param to trigger retry if any of the requests failed (getBookieInfoRetryIntervalSeconds). We'll only retry the ones that actually failed (along with any new additions since the last run). This is useful because bookie startup triggers the addition of the bookie node to zk before the bookie actually becomes available for the bookie info request, so there can be rare races in the unit tests between BookieInfoReader requesting the info and the bookie actually being up. Also, add a method to allow tests to wait for updates to be reflected. PerChannelBookieClient: fix error handling for BookieInfo keys Passing a key corresponding to a GET_BOOKIE_INFO operation to errorOutReadKey results in a casting exception, clean up the invalid calls. BookKeeperClusterTestCase: add killBookieAndWaitForZK Should reduce the need for tests to wait for an arbitrary period to let the cluster "settle". BookKeeperDiskSpaceWeightedLedgerPlacementTest: This test was heavily time dependent, and the Thread.sleep values did not work universally. Instead, eliminate the arbitrary Thread.sleep values and instead verify the free space changes on each change. Also, switch the delay on testDiskSpaceWeightedBookieSelectionWithPeriodicBookieInfoUpdate to simply use an atomic boolean to signal the value switch. Signed-off-by: Samuel Just <sjust@salesforce.com> Author: Samuel Just <sjust@salesforce.com> Reviewers: Enrico Olivelli <eolivelli@gmail.com> This patch had conflicts when merged, resolved by Committer: Sijie Guo <sijie@apache.org> This closes #275 from athanatos/forupstream/BOOKKEEPER-1102 {noformat}
2. FAILURE: Integrated in Jenkins build bookkeeper-master #1836 (See [<https://builds.apache.org/job/bookkeeper-master/1836/>]) BOOKKEEPER-1102: Clarify BookieInfoReader and fix associated test (sijie: [<https://github.com/apache/bookkeeper/commit/a5f8580f53464065243a9af038935f5893434166>]) * (edit) bookkeeper-server/src/test/java/org/apache/bookkeeper/client/BookKeeperDiskSpaceWeightedLedgerPlacementTest.java * (edit) bookkeeper-server/src/main/java/org/apache/bookkeeper/conf/ClientConfiguration.java * (edit) bookkeeper-server/src/main/java/org/apache/bookkeeper/proto/PerChannelBookieClient.java * (edit) bookkeeper-server/src/main/java/org/apache/bookkeeper/client/BookieInfoReader.java * (edit) bookkeeper-server/src/test/java/org/apache/bookkeeper/test/BookKeeperClusterTestCase.java