Item 332
git_comments:

1. * * Interface to provide XceiverClient when needed.
2. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
3. .test(d.navigator&&d.navigator.userAgent);return{log:e("log"),info:e("info"),warn:e("warn"),error:e("error"),debug:function(){var c=e("debug");return function(){a&&c.apply(b,arguments)}}()}}]}function Tg(a){return a+""}function Ug(a,b){return"undefined"!==typeof a?a:b}function Gd(a,b){return"undefined"===typeof a?b:"undefined"===typeof b?a:a+b}function Vg(a,b){switch(a.type){case q.MemberExpression:if(a.computed)return!1;break;case q.UnaryExpression:return 1;case q.BinaryExpression:return"+"!==
4. AngularJS v1.8.0 (c) 2010-2020 Google, Inc. http://angularjs.org License: MIT
5. g,"%2F"));d=c.join("/");b.$$path=d;b.$$search=hc(a.search);b.$$hash=decodeURIComponent(a.hash);b.$$path&&"/"!==b.$$path.charAt(0)&&(b.$$path="/"+b.$$path)}function yc(a,b){return a.slice(0,b.length)===b}function ya(a,b){if(yc(b,a))return b.substr(a.length)}function Da(a){var b=a.indexOf("#");return-1===b?a:
6. After restarting SCM, ensure the sequenceId for the container is the same as before.
7. Create some keys to write data into the open containers
8. Ensure 3 replicas are reported successfully as expected.
9. * * Integration test to ensure a container can be closed and its replicas * reported back correctly after a SCM restart.
10. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
11. Pick any container on the cluster, get its pipeline, close it and then wait for the container to close
12. * * Retrieves the containerReplica set for a given container or fails the test * if the container cannot be found. This is a helper method to allow the * container replica count to be checked in a lambda expression. * @param c The container for which to retrieve replicas * @return
13. Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
14. * * For when the request is not successful. * For a successful request, the other constructor should be used.
15. * * Base class for responses that need to move keys from an arbitrary table to * the deleted table.
16. **comment:** For OmResponse with failure, this should do nothing. This method is not called in failure scenario in OM code.
    label: code-design
17. * * Adds the operation of deleting the {@code keyName omKeyInfo} pair from * {@code fromTable} to the batch operation {@code batchOperation}. The * batch operation is not committed, so no changes are persisted to disk. * The log transaction index used will be retrieved by calling * {@link OmKeyInfo#getUpdateID} on {@code omKeyInfo}.
18. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * <p> * http://www.apache.org/licenses/LICENSE-2.0 * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
19. * * Handles requests to move open keys from the open key table to the delete * table. Modifies the open key table cache only, and no underlying databases. * The delete table cache does not need to be modified since it is not used * for client response validation.
20. Update table cache.
21. If an open key is no longer present in the table, it was committed and should not be deleted.
22. For each bucket where keys will be deleted from, get its bucket lock and update the cache accordingly.
23. **comment:** No need to add cache entries to delete table. As delete table will be used by DeleteKeyService only, not used for any client response validation, so we don't need to add to cache.
    label: code-design
24. Set the UpdateID to current transactionLogIndex
25. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * <p> * http://www.apache.org/licenses/LICENSE-2.0 * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
26. * * Handles responses to move open keys from the open key table to the delete * table. Modifies the open key table and delete table databases.
27. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * <p> * http://www.apache.org/licenses/LICENSE-2.0 * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
28. * * For when the request is not successful. * For a successful request, the other constructor should be used.
29. * * Tests OMOpenKeysDeleteRequest.
30. * * Adds {@code openKeys} to the open key table DB only, and asserts that they * are present after the addition. * @throws Exception
31. * * Constructs a list of {@link OpenKeyBucket} objects of size {@code numKeys}. * The keys created will all have the same volume and bucket, but * randomized key names and client IDs. These keys are not added to the * open key table. * * @param volume The volume all open

keys created will have. * @param bucket The bucket all open keys created will have. * @param numKeys The number of keys with randomized key names and client * IDs to create. * @return A list of new open keys with size {@code numKeys}.

32. * * Expands all the open keys represented by {@code openKeyBuckets} to their * full * key names as strings. * @param openKeyBuckets * @return
33. * * Creates an {@code OpenKeyDeleteRequest} to delete the keys represented by * {@code keysToDelete}. Returns an {@code OMRequest} which encapsulates this * {@code OpenKeyDeleteRequest}.
34. * * Adds {@code openKeys} to the open key table DB only, and asserts that they * are present after the addition. Adds each key to the table with a single * block of size {@code keySize}. * @throws Exception
35. * * Tests removing keys from the open key table cache that never existed there. * The operation should complete without errors. * <p> * This simulates a run of the open key cleanup service where a set of * expired open keys are identified and passed to the request, but before * the request can process them, those keys are committed and removed from * the open key table. * @throws Exception
36. * * Runs the validate and update cache step of * {@link OMOpenKeysDeleteRequest} to mark the keys in {@code openKeys} * as deleted in the open key table cache. * Asserts that the call's response status is {@link Status#OK}. * @throws Exception
37. * * Tests removing keys from the open key table cache that have the same * name, but different client IDs. * @throws Exception
38. * * Constructs a list of {@link OpenKey} objects of size {@code numKeys}. * The keys created will all have the same volume, bucket, and * key names, but randomized client IDs. These keys are not added to the * open key table. * * @param volume The volume all open keys created will have. * @param bucket The bucket all open keys created will have. * @param key The key name all open keys created will have. * @param numKeys The number of keys with randomized key names and client * IDs to create. * @return A list of new open keys with size {@code numKeys}.
39. * * Constructs a new {@link OMOpenKeysDeleteRequest} objects, and calls its * {@link OMOpenKeysDeleteRequest#preExecute} method with {@code * originalOMRequest}. It verifies that {@code originalOMRequest} is modified * after the call, and returns it. * @throws Exception
40. * * Tests adding multiple keys to the open key table, and updating the table * cache to only remove some of them. * Keys not removed should still be present in the open key table. * Mixes which keys will be kept and deleted among different volumes and * buckets. * @throws Exception
41. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * <p> * http://www.apache.org/licenses/LICENSE-2.0 * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
42. * * Tests metrics set by {@link OMOpenKeysDeleteRequest}. * Submits a set of keys for deletion where only some of the keys actually * exist in the open key table, and asserts that the metrics count keys * that were submitted for deletion versus those that were actually deleted. * @throws Exception
43. Will not be equal, as UserInfo will be set.
44. * * Tests deleting a subset of keys from the open key table DB when the keys * have associated block data.
45. These keys should not have been removed from the open key table.
46. * * Tests OMOpenKeysDeleteResponse.
47. These keys should have been moved from the open key table to the delete table.
48. open keys with no associated block data should have been removed from the open key table, but not added to the deleted table.
49. * * Tests deleting a subset of keys from the open key table DB when the keys * have no associated block data.
50. Add to the open key table DB, not cache. In a real execution, the open key would have been removed from the cache by the request, and it would only remain in the DB.
51. * * Tests attempting deleting keys from the open key table DB when the * submitted response has an error status. In this case, no changes to the * DB should be made.
52. Operations are only added to the batch by this method when status is OK.
53. * * Creates {@code numKeys} open keys with random names, maps each one to a * new {@link OmKeyInfo} object, adds them to the open key table cache, and * returns them. These keys will have no associated block data.
54. If an error occurs in the response, the batch operation moving keys from the open key table to the delete table should not be committed.
55. If status is not OK, this will do nothing.
56. * * Constructs an {@link OMOpenKeysDeleteResponse} to delete the keys in * {@code keysToDelete}, with the completion status set to {@code status}. * If {@code status} is {@link Status#OK}, the keys to delete will be added * to a batch operation and committed to the database. * @throws Exception
57. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * <p> * http://www.apache.org/licenses/LICENSE-2.0 * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
58. * * Creates {@code numKeys} open keys with random names, maps each one to a * new {@link OmKeyInfo} object, adds them to the open key table cache, and * returns them. * If {@code keyLength} is greater than 0, adds one block with that many * bytes of data for each key. * @throws Exception
59. These keys should not have been moved out of the open key table.
60. * * Drop a column Family/Table in db.
61. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
62. * * RocksDB specific utility functions.
63. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
64. * Never constructed. *

65. * * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * <p> * http://www.apache.org/licenses/LICENSE-2.0 * <p> * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
66. * * Tests the ozone getconf command.
67. Test volume quota. Test volume quota: write key. Test volume quota: write file. Test volume quota: write key(with two blocks), test allocateBlock fails. Test volume quota: write large key(with five blocks), the first four blocks will succeed，while the later block will fail. Test bucket quota. Set quota In Bytes for a smaller value
68. Test bucket quota: write key. The remaining quota does not satisfy a block size, so the write fails.
69. Write failed, bucket usedBytes should be 0
70. Test bucket quota: write large key(with five blocks), the first four blocks will succeed，while the later block will fail.
71. AllocateBlock failed, bucket usedBytes should be 4 * blockSize
72. Test bucket quota: write file. The remaining quota does not satisfy a block size, so the write fails.
73. As second time with same client id and call id, this request should not be executed ratis server should return from cache.
74. check bucket and volume quota
75. check bucket and volume quota
76. check bucket and volume quota
77. * * @return the number of bytes used by blocks pointed to by {@code omKeyInfo}.
78. * * Check bucket quota in bytes. * @param omBucketInfo * @param allocateSize * @throws IOException
79. * * Adds one block to {@code keyInfo} with the provided size and offset.
80. **comment:** ozKey had no block information associated with it, so it should have been removed from the key table but not added to the delete table.
    label: code-design

git_commits:

1. **summary:** Merge master into HDDS-1880-Decom
   message: Merge master into HDDS-1880-Decom

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Decommissioning and maintenance mode in Ozone
   description: This is the umbrella jira for decommissioning support in Ozone. Design doc will be attached soon.

jira_issues_comments:

1. Is this feature still targeted for 0.6.0?
2. [~arp] Nobody is actively working on this at the moment so it will not make 0.6.0.
3. I am managing the 1.1.0 release and we currently have more than 600 issues targeted for 1.1.0. I am moving the target field to 1.2.0. If you are actively working on this jira and believe this should be targeted to 1.1.0 release, Please change the target field back to 1.1.0 before Feb 05, 2021.
4. In current implements, we don't allow nodes decommission if they are in maintenance state. There is the following scenario in production environment. Based on HDFS experience, if we have a large cluster and we want to decommission many nodes. Workload will be high if we decommission lots of nodes at the same time, so we will decommission nodes many times and decommission tens of nodes at a time. We expect that Other nodes which we want to decommission are readable but not writable during some nodes in decommissioning state.   So here suggest that allow it to decommission if one node in maintenance state. In current implement, we must do that: 1) set maintenance;2)recommission;3)set decommision. With my suggestion, just do that: 1) set maintenance; 2) set decommision Does it make sense? [~sodonnell] [~elek]
5. **body:** [~ferhui] Thanks for the question. Putting a node to maintenance may not be "free". It is possible there will need to be some replication to ensure the containers are still minimally replicated (2 copies by default, but can be set to 1), so putting a lot of nodes to maintenance at once could be expensive too. What I would really like to see, is that an operator should not need to worry about the number of nodes they take out of service at the same time. For example, if you have a 100 node cluster and want to decommission 50 nodes, and assuming: 1. The load on the cluster is such that 50 nodes can handle it 2. There is less than 50% space used on the cluster You should just decommission all 50 nodes at once and the system should take care to throttle the replication so it remains functional. In some respects, Ozone should handle this better than HDFS, as it is replicating at the container level. The total containers on the cluster should be much less than the number of blocks on a HDFS cluster, so there are less replication jobs to schedule and keep track of, However the current replication manager implementation in Ozone will not handle the above scenario well. It will schedule all the replication jobs out to all the DNs on the first pass. While the tasks are queued, there are timeouts when they get reschedule etc. If our goal is to be able to decommission a large part of the cluster at once, then I think we need some work in the replication manager, but it should be possible to achieve this safely without decommissioning in small batches.
   label: code-design