

git_comments:

1. 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.
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. 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.
4. 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.

git_commits:

1. **summary:** Add Helix Distributed lock module (#673)
message: Add Helix Distributed lock module (#673) Add Helix Lock module.

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

1. **title:** Add Helix Distributed lock module
body: #### Issues - [X] My PR addresses the following Helix issues and references them in the PR description: (#672)(<https://github.com/apache/helix/issues/672>) #### Description - [X] Here are some details about my PR, including screenshots of any UI changes: For distributed systems that use Helix, there is a common request for a mechanism to make sure different processes can cooperate correctly. For example, if two processes that run on different machines, or different networks, or even different data centers would like to work on the same resource, we must have a mutually exclusive lock to secure their operations do not step on each other. Only when a process acquires the lock, it can perform the operation. Therefore, we want to implement a distributed lock in Helix to benefit clients. This PR adds the new Helix distributed lock module in Helix repo. #### Tests - [X] The following is the result of the "mvn test" command on the appropriate module: Waiting for time window to expire END TestParticipantMonitor START test at Mon Jan 06 12:19:14 PST 2020 START TestZKPathDataDumpTask_test at Mon Jan 06 12:19:14 PST 2020 END TestZKPathDataDumpTask_test at Mon Jan 06 12:19:14 PST 2020 END test at Mon Jan 06 12:19:14 PST 2020, took: 204ms. START testCapacityReached at Mon Jan 06 12:19:14 PST 2020 START TestZKPathDataDumpTask_testCapacityReached at Mon Jan 06 12:19:14 PST 2020 END testCapacityReached at Mon Jan 06 12:19:14 PST 2020, took: 221ms. START testBasic at Mon Jan 06 12:19:20 PST 2020 START TestHelixConfigAccessor_testBasic at Mon Jan 06 12:19:20 PST 2020 END TestHelixConfigAccessor_testBasic at Mon Jan 06 12:19:20 PST 2020 END testBasic at Mon Jan 06 12:19:20 PST 2020, took: 292ms. START testSetNonexistentParticipantConfig at Mon Jan 06 12:19:20 PST 2020 START TestHelixConfigAccessor_testSetNonexistentParticipantConfig at Mon Jan 06 12:19:20 PST 2020 3251077 [TestNGInvoker-testSetNonexistentParticipantConfig()] ERROR org.apache.helix.manager.zk.ZKUtil - Invalid instance setup, missing znode path:

```

/TestHelixConfigAccessor_testSetNonexistentParticipantConfig/CONFIGS/PARTICIPANT/localhost_12918
3251092 [TestNGInvoker-testSetNonexistentParticipantConfig()] ERROR org.apache.helix.manager.zk.ZKUtil -
Invalid instance setup, missing znode path:
/TestHelixConfigAccessor_testSetNonexistentParticipantConfig/INSTANCES/localhost_12918/MESSAGES
3251092 [TestNGInvoker-testSetNonexistentParticipantConfig()] ERROR org.apache.helix.manager.zk.ZKUtil -
Invalid instance setup, missing znode path:
/TestHelixConfigAccessor_testSetNonexistentParticipantConfig/INSTANCES/localhost_12918/CURRENTSTATES
3251092 [TestNGInvoker-testSetNonexistentParticipantConfig()] ERROR org.apache.helix.manager.zk.ZKUtil -
Invalid instance setup, missing znode path:
/TestHelixConfigAccessor_testSetNonexistentParticipantConfig/INSTANCES/localhost_12918/STATUSUPDATES
3251093 [TestNGInvoker-testSetNonexistentParticipantConfig()] ERROR org.apache.helix.manager.zk.ZKUtil -
Invalid instance setup, missing znode path:
/TestHelixConfigAccessor_testSetNonexistentParticipantConfig/INSTANCES/localhost_12918/ERRORS END
TestHelixConfigAccessor_testSetNonexistentParticipantConfig at Mon Jan 06 12:19:20 PST 2020 END
testSetNonexistentParticipantConfig at Mon Jan 06 12:19:20 PST 2020, took: 110ms. Shut down zookeeper at port
2183 in thread main [INFO] Tests run: 890, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3,264.509 s - in
TestSuite [INFO] [INFO] Results: [INFO] [INFO] Tests run: 890, Failures: 0, Errors: 0, Skipped: 0 [INFO] [INFO]
----- [INFO] BUILD SUCCESS [INFO] -----
----- [INFO] Total time: 54:28 min [INFO] Finished at: 2020-01-
06T12:19:26-08:00 [INFO] -----
### Commits - [X] My
commits all reference appropriate Apache Helix GitHub issues in their subject lines, and I have squashed multiple
commits if they address the same issue. In addition, my commits follow the guidelines from "[How to write a good
git commit message](http://chris.beams.io/posts/git-commit/)": 1. Subject is separated from body by a blank line 1.
Subject is limited to 50 characters (not including Jira issue reference) 1. Subject does not end with a period 1.
Subject uses the imperative mood ("add", not "adding") 1. Body wraps at 72 characters 1. Body explains "what"
and "why", not "how" #### Code Quality - [X] My diff has been formatted using helix-style.xml

```

github_pulls_comments:

1. What is this test result from? Did you run the test on the helix-lock module or helix-core?
2. > What is this test result from? Did you run the test on the helix-lock module or helix-core? Yeah, this is helix-core. Just wanted to show that the new module doesn't break it. But you're right. It doesn't test new module. We will have test when we add implementation code in lock module.
3. This PR is ready to be merged, approved by @jiajunwang Final commit message "Add Helix Lock module"

github_pulls_reviews:

1. Do you need all these?
2. Same for this file. Remove all the components, which are unnecessary. Otherwise, the file is too large and imports are not required.
3. I removed some that won't be needed for sure.
4. I removed some that won't be needed for sure.
5. These dependencies are the same in the parent pom. I think it would be more clear and easier to maintain if you remove these dependencies in this child pom, because this child pom file will inherit all properties and dependencies from parent POM. Unless you want to use a different version/additional dependency. Same with following duplicate dependencies.
6. I actually don't find this one in parent pom. Do you have another example that is in the parent?
7. Oh, it is not `snakeyaml`. I actually selected a block but github doesn't show it. I meant this part as an example: <https://github.com/apache/helix/blob/d6ba1f3e792651e250d60b3df1fc6b537934f69e/pom.xml#L404-L413> Do you think the dependencies could be removed?
8. yeah, removed.
9. I see that the parent pom also has jetty-server dependency: `` <dependency>
<groupId>org.eclipse.jetty</groupId> <artifactId>jetty-server</artifactId> <version>8.1.8.v20121106</version>
</dependency> `` Is there any reason you want to use this specific version(9.1.0.RC0)? To me, 8.1.8.v20121106 is pretty old. Not sure if it is time to upgrade this dependency in the parent pom, too.
10. I think it's safer to keep parent pom the same as before. Not quite sure about its impact.
11. Nit, I understand the other modules' names are also hardcoded. I would just make a variable
`module_name="helix-lock"` (Or better, can read from pom file) and reuse the module. I've seen duplicated code in this script, which a for loop could handle. Maybe we can have another PR to clean this script.
12. Nit, DEBUG and WARN? What level is expected?
13. My point is, you are using an RC version 9.1.0.RC0 here for jetty-server. RC version is a release candidate. If you have any specific reason to use another version than 8.1.8.v20121106 in parent, you may want to choose a release version. What do you think?

14. Question: My understanding is that this distributed lock is supposed to be a general module that does not assume the use of Helix. Should we call this something like zk-distributed-lock?
15. Are we planning on adding ivy files? Otherwise, this won't do anything...
16. Yep, agree. Let's do a cleanup in a separate PR.
17. I'm actually thinking that it is a Helix lock, and with different kinds of implementations. For example, we implement a lock based on zookeeper now, and we may implement a different version based on etcd later.
18. yeah, ivy files are necessary when we build release internally. Otherwise it won't work.
19. helix-rest -> helix-lock
20. Nit, This timestamp should be newer, I guess.

jira_issues:

jira_issues_comments: