Item 320
git_comments:

1. * * Interface implemented by all Collection API commands. Collection API commands are defined in classes whose names ends in {@code Cmd}.
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. * * This class contains "smaller" Collection API commands implementation as well as the interface implemented by all commands. * Previously these implementations in {@link OverseerCollectionMessageHandler} were relying on methods implementing the * functional interface.
4. * * admin path passed to Overseer is a constant, including in tests.
5. * * Method used by Per Replica States implementation to force the cluster state updater to immediately reload a collection from Zookeeper. * This method is not used when cluster state updates are distributed.
6. * * Data passed to Collection API command execution, to allow calls from either the {@link OverseerCollectionMessageHandler} * when commands are executed on the Overseer, or - in a future change - allow Collection API commands to be executed in a * distributed way, unrelated to and not depending upon Overseer abstractions (including overseer collection message handling).
7. * * Command delegating to Overseer to retrieve cluster state update stats from a Collection API call. This does not make * sense when cluster state updates are distributed given Overseer does not see them and can't collect stats.
8. * * This method enables the commands to enqueue to the overseer cluster state update. This should only be used when the command * is running in the Overseer (and will throw an exception if called when Collection API is distributed)
9. * 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.
10. * backward compatibility reasons, add the response with the async ID as top level. * This can be removed in Solr 9
11. * 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.
12. * * This class contains helper methods used by commands of the Collection API. Previously these methods were in * {@link OverseerCollectionMessageHandler} and were refactored out to (eventually) allow Collection API commands to be * executed outside the context of the Overseer.
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. * * Context passed to Collection API commands when they execute in the Overseer.
15. TODO: consider doing this once after the loop for all replicas rather than writing state.json repeatedly
16. Distributed updates don't need to do anything for PRS collections that wrote state.json directly For non PRS collections, distributed updates have to be executed if that's how the cluster is configured
17. PRS collections updated ZK state.json in the loop above. When Overseer is managing cluster state updates, need to tell it to refresh itself to know about the replicas and be able to execute nodes shard requests regarding the replicas.
18. When cluster state updates are handled by Overseer, ask it to load that collection it doesn't know about. When cluster state updates are distributed, ZK is the source of truth for all nodes so no reload needed.
19. take advantage of this through method Overseer.submit() accessed via CollectionCommandContext.submitIntraProcessMessage()). stats below do not make sense when cluster state updates are distributed. Return them empty.

git_commits:

1. **summary:** SOLR-15157: refactor Collection API to separate from Overseer and message handling abstractions (#2390)
   message: SOLR-15157: refactor Collection API to separate from Overseer and message handling abstractions (#2390) No functional changes. In preparation of distributing the Collection API command execution.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** SOLR-15157: refactor Collection API to separate from Overseer and message handling abstractions
   body: This refactoring is inserting a layer of abstraction (`CollectionCommandContext`) between Collection API commands and the `OverseerCollectionMessageHandler`, to enable future changes where Collection API messages are executed outside of Overseer. There are (almost) no other changes in this PR. Two exceptions: 1. in `CreateCollectionCmd`, appropriate conditions were added around recently added calls related to Per Replica States collections and 2. a minor fix in test `OverseerStatusTest` regarding distributed updates and Overseer stats. I'd rather this PR not linger around for too long if possible. Given `OverseerCollectionMessageHandler` has lost most of its contents, merging in new changes is a highly manual process. This refactoring is intended to eventually enable constructions such as shown in the PoC (based on a slightly older snapshot of master). See line [348 in CollectionsHandler](https://github.com/murblanc/lucene-solr/commit/10cdad0e2618ea82b2b97f4a78cc0f3ff8df8082#diff-582348d44491dcb0ce1dfb169fb544e9e95620b2d0448eb1a1744f4e8dd5a349R348) there, when a request is received to create a collection, rather than enqueue a ZK message to the Collection API queue for Overseer, the execution is done locally in method [distributedCollectionCreation](https://github.com/murblanc/lucene-solr/commit/10cdad0e2618ea82b2b97f4a78cc0f3ff8df8082#diff-582348d44491dcb0ce1dfb169fb544e9e95620b2d0448eb1a1744f4e8dd5a349R280). This method needs to call the Collection API command, and does not execute in the context of the Overseer. The motivation for doing the refactoring separately (as discussed in [Slack](https://the-asf.slack.com/archives/CEKUCUNE9/p1612908679186300)) is to keep the hardest to review part (code moving around) clean of any other changes (and a side benefit is that the actual Collection API distribution will take quite some time, and having part of it be such a refactoring waiting for a few weeks on a branch is a recipe for not being able to merge it safely).

github_pulls_comments:

1. @madrob @chatman @gerlowskija @sigram @noblepaul - if you want to have a look. You've all recently touched parts of the code and/or are familiar with it.
2. I have done my best to preserve the order of code originating in `OverseerCollectionMessageHandler` as it moved to `CollectionHandlingUtils` and `CollApiCmds` to make an eyeball diff/comparison _slightly_ easier...

github_pulls_reviews:

1. *PREDICTABLE_RANDOM:* This random generator (java.util.Random) is predictable [(details)](https://find-sec-bugs.github.io/bugs.htm#PREDICTABLE_RANDOM)

jira_issues:

jira_issues_comments: