

git_comments:

1. * * The evaluator has only contacted the driver after the expiration period.
2. * * The evaluator has had its context/running task processed.
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. * * @return true if the transition of {@link EvaluatorRestartState} is legal.
5. * * @return true if the evaluator has heartbeated back to the driver.
6. * * The evaluator has had its recovery heartbeat processed.
7. * * The state that the evaluator is in in the driver restart process.
8. * * The evaluator is not a restarted instance. Not expecting.
9. * * Received the evaluator heartbeat, but have not yet processed it.
10. * * Have not yet heard back from an evaluator, but we are expecting it to report back.
11. **comment:** set the state machine in motion. TODO[REEF-560]: Call onDriverRestartCompleted() (to do in REEF-617) on a Timer.
label: requirement
12. * * Signals to the {@link DriverRestartManager} that an evaluator is to be expected to report back after restart.
13. * * Signals to the {@link DriverRestartManager} that an evaluator has reported back after restart.
14. * * Signals to the {@link DriverRestartManager} that an evaluator has had its recovery heartbeat processed.
15. * * Signals to the {@link DriverRestartManager} that an evaluator has had its running task processed.
16. * * Signals to the {@link DriverRestartManager} that an expected evaluator has been expired.
17. * * the negation of {@link #hasRestarted()}.
18. **comment:** TODO[REEF-617]: Create EvaluatorManager for recovered evaluator and call onEvaluatorHeartbeatMessage().
label: code-design
19. **comment:** TODO[REEF-617]: Create EvaluatorManager for expired evaluator and close it.
label: code-design
20. * * Process an evaluator heartbeat message.
21. * First message from a running evaluator. The evaluator can be a new evaluator or be a previous evaluator * from a separate application attempt. In the case of a previous evaluator, if the restart period has not * yet expired, we should register it and trigger context active and task events. If the restart period has * expired, we should return immediately after setting its remote ID in order to close it.
22. **comment:** TODO[REEF-617]: Move restart completion logic to DriverRestartManager.
label: code-design
23. **comment:** TODO[REEF-617]: Move evaluator recovery to EvaluatorHeartbeatHandler and reregister evaluator here.
label: code-design
24. Don't do anything if evaluator has expired. Close it immediately upon exit of this method.

git_commits:

1. **summary:** [REEF-616] Keep state of previous evaluators with a state machine
message: [REEF-616] Keep state of previous evaluators with a state machine This switches to a state machine model for evaluator statuses for driver restart. JIRA: [REEF-616]
(<https://issues.apache.org/jira/browse/REEF-616>) Pull Request: This closes #396

github_issues:**github_issues_comments:**

github_pulls:

- [illegible]

19. **title:** [REEF-616] Keep state of previous evaluators with a state machine
body: This addressed the issue by - Switching to a state machine model for evaluator statuses for driver restart. JIRA: [REEF-616](https://issues.apache.org/jira/browse/REEF-616)

github_pulls_comments:

1. [Reef-pull-request-ubuntu #367](https://builds.apache.org/job/Reef-pull-request-ubuntu/367/) SUCCESS
This pull request looks good
2. [REEF-pull-request-windows3 #274](https://builds.apache.org/job/REEF-pull-request-windows3/274/) SUCCESS
This pull request looks good
3. I'll do a pass.
4. @afchung I've done a pass.
5. @markusweimer Addressed comments. Please have another look. Thanks!
6. [REEF-pull-request-windows3 #278](https://builds.apache.org/job/REEF-pull-request-windows3/278/) SUCCESS
This pull request looks good
7. **body:** Looks good, I will test & merge.
label: test
8. [Reef-pull-request-ubuntu #371](https://builds.apache.org/job/Reef-pull-request-ubuntu/371/) FAILURE
Looks like there's a problem with this pull request

github_pulls_reviews:

1. This, as well as the assignment to `state`, could move to the attribute declaration.
2. It might make sense to code the negation into a method, e.g. `state.hasNotRestarted()`
3. This can be `private`, as it is only called within this class.
4. This could be a `private` method of its own. e.g. `haveAllExpectedEvaluatorsReported()`.
5. **body:** There are some methods in this class which are not used (any more?): - `setEvaluatorExpired` - `setEvaluatorReregistered` - `isRestartCompleted` - `hasRestarted` Will they be used in future commits?
label: code-design
6. Maybe add a static method `boolean isLegalTransition(EvaluatorRestartState old, EvaluatorRestartState new)` to check transitions between states.
7. Other than `isRestartCompleted`, the rest of the methods will be used in future commits.
8. **body:** I preferred keeping it public, since some of the other `setEvaluator` calls are needed to be called from the outside, and left this here to keep it consistent. What do you think?
label: code-design
9. OK, keep it `public`

jira_issues:

1. **summary:** Keep state of previous evaluators with a state machine
description: Right now it's using two separate sets. This will be much more complicated when the concept of evaluator expiration is factored in. The more concise pattern here to use is the state machine.
2. **summary:** Keep state of previous evaluators with a state machine
description: Right now it's using two separate sets. This will be much more complicated when the concept of evaluator expiration is factored in. The more concise pattern here to use is the state machine.
label: code-design
3. **summary:** Keep state of previous evaluators with a state machine
description: Right now it's using two separate sets. This will be much more complicated when the concept of evaluator expiration is factored in. The more concise pattern here to use is the state machine.

jira_issues_comments:

1. Implemented via [#396](https://github.com/apache/incubator-reef/pull/396)