Item 272
git_comments:

git_commits:

1. **summary:** [AIRFLOW-2703] Catch transient DB exceptions from scheduler's heartbeat it does not crash (#3650)
   message: [AIRFLOW-2703] Catch transient DB exceptions from scheduler's heartbeat it does not crash (#3650) If there is any issue in DB connection then rest of the functions take care of those exceptions but in heartbeat of scheduler, there is no handling for this kind of situation. Airflow Scheduler should not crash if a "transient" DB exception occurs in the heartbeat of scheduler.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [AIRFLOW-2703] exceptions from scheduler's heartbeat is handled so that scheduler does not crash
   body: If there is any issue in DB connection then rest of the functions take care of those exceptions but in heartbeat of scheduler, there is no handling for this kind of situation. Airflow Scheduler should not crash if any exception occurs in the heartbeat of scheduler. Make sure you have checked all steps below. ### JIRA - [x] My PR addresses the following [Airflow JIRA] (https://issues.apache.org/jira/browse/AIRFLOW-2703) issues and references them in the PR title. - https://issues.apache.org/jira/browse/AIRFLOW-2703 ### Description - [x] Here are some details about my PR, including screenshots of any UI changes: ### Tests - [x] My PR adds the following unit tests __OR__ does not need testing for this extremely good reason: no new test case ### Commits - [x] My commits all reference JIRA 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 2. Subject is limited to 50 characters 3. Subject does not end with a period 4. Subject uses the imperative mood ("add", not "adding") 5. Body wraps at 72 characters 6. Body explains "what" and "why", not "how" ### Documentation - [x] In case of new functionality, my PR adds documentation that describes how to use it. - When adding new operators/hooks/sensors, the autoclass documentation generation needs to be added. ### Code Quality - [x] Passes `git diff upstream/master -u -- "*.py" | flake8 --diff`

github_pulls_comments:

1. # [Codecov](https://codecov.io/gh/apache/incubator-airflow/pull/3650?src=pr&el=h1) Report > Merging [#3650](https://codecov.io/gh/apache/incubator-airflow/pull/3650?src=pr&el=desc) into [master] (https://codecov.io/gh/apache/incubator-airflow/commit/b8be322d3badfeadfa8f08e0bf92a12a6cd26418?src=pr&el=desc) will **increase** coverage by `1.87%`. > The diff coverage is `85%`. [![Impacted file tree graph](https://codecov.io/gh/apache/incubator-airflow/pull/3650/graphs/tree.svg?width=650&token=WdLKlKHOAU&height=150&src=pr)](https://codecov.io/gh/apache/incubator-airflow/pull/3650?src=pr&el=tree) ```diff @@ Coverage Diff @@ ## master #3650 +/- ## == + Coverage 75.79% 77.67% +1.87% == Files 199 204 +5 Lines 15946 15849 -97 == + Hits 12086 12310 +224 + Misses 3860 3539 -321 ``` | [Impacted Files](https://codecov.io/gh/apache/incubator-airflow/pull/3650?src=pr&el=tree) | Coverage Δ | | |---|---|---| | [airflow/jobs.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy9qb2JzLnB5) | `82.63% <85%> (+0.23%)` | :arrow_up: | | [airflow/sensors/s3_key_sensor.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy9zZW5zb3JzL3MzX2tleV9zZW5zb3IucHk=) | `31.03% <0%> (-68.97%)` | :arrow_down: | | [airflow/sensors/s3_prefix_sensor.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy9zZW5zb3JzL3MzX3ByZWZpeF9zZW5zb3IucHk=) | `41.17% <0%> (-58.83%)` | :arrow_down: | | [airflow/utils/helpers.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy91dGlscy9oZWxwZXJzLnB5) | `71.34% <0%> (-13.04%)` |

:arrow_down: | | [airflow/hooks/mysql_hook.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy9ob29rcy9teXNxbF9ob29rLnB5) | `78% <0%> (-12%)` | :arrow_down: | | [airflow/sensors/sql_sensor.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy9zZW5zb3JzL3NxbF9zZW5zb3IucHk=) | `90.47% <0%> (-9.53%)` | :arrow_down: | | [airflow/configuration.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy9jb25maWd1cmF0aW9uLnB5) | `84.07% <0%> (-5.19%)` | :arrow_down: | | [airflow/utils/state.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy91dGlscy9zdGF0ZS5weQ==) | `93.33% <0%> (-3.34%)` | :arrow_down: | | [airflow/models.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy9tb2RlbHMucHk=) | `88.78% <0%> (-2.93%)` | :arrow_down: | | [airflow/utils/email.py](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree#diff-YWlyZmxvdy91dGlscy9lbWFpbC5weQ==) | `97.4% <0%> (-2.6%)` | :arrow_down: | | ... and [61 more](https://codecov.io/gh/apache/incubator-airflow/pull/3650/diff?src=pr&el=tree-more) | | ------ [Continue to review full report at Codecov](https://codecov.io/gh/apache/incubator-airflow/pull/3650?src=pr&el=continue). > **Legend** - [Click here to learn more](https://docs.codecov.io/docs/codecov-delta) > `Δ = absolute <relative> (impact)`, `ø = not affected`, `? = missing data` > Powered by [Codecov](https://codecov.io/gh/apache/incubator-airflow/pull/3650?src=pr&el=footer). Last update [b8be322...644f6bb](https://codecov.io/gh/apache/incubator-airflow/pull/3650?src=pr&el=lastupdated). Read the [comment docs](https://docs.codecov.io/docs/pull-request-comments).
2. @ashb can you please review this?
3. What sort of exceptions happen here? Do we need to worry about the same error happening over-and-over again in quick succession producing gigabytes of logs very quickly? (I've seen this happen with other software) My initial reaction is that knowing that there is a problem by the scheduler crashing is a useful thing, and that if you want it to restart that's the job of an external process supervisor. Open to a reason otherwise though.
4. @ashb I changed the exception handling to catch only for that which occurs due to lost connection to MySQL server.
5. @ashb please review the change.
6. @feng-tao can you please review the changes?
7. @ashb @feng-tao is there any issue with this PR that I need to change, if not we can go ahead and merge this. This is blocking one of our production issue.
8. My major concern is that this would hide serious problems with the connection to the DB (say the DB crashed and isn't running anymore, or the tables are corrupt) - this would then end up creating lots of logs but just keep trying endlessly. Maybe it's worth only allowing _n_ failures in a 60s window, or do you think I am worrying needlessly?
9. @ashb thanks for replying. Our approach was to take care of the intermittent DB connection loss due to which scheduler crashes. If there is any continuous issue with the connection to the DB then we were expecting the external process supervisor to take care of it and restart the scheduler. Does this sound correct?
10. > If there is any continuous issue with the connection to the DB then we were expecting the external process supervisor to take care of it and restart the scheduler. Does that throw up as a different exception class? I guess my question is does this change now mask that issue and prevent the scheduler from failing over in the first case?
11. @ashb As we are handling only OperationalError so this will handle only "Exception raised for errors that are related to the database's operation and not necessarily under the control of the programmer, e.g. an unexpected disconnect occurs, the data source name is not found, a transaction could not be processed, a memory allocation error occurred during processing, etc." we think that these intermittent issues should not bring the whole scheduler down. Any other DB related issue will bring the scheduler down as they will come under different class . |__Error |__InterfaceError |__DatabaseError |__DataError |___OperationalError_ |__IntegrityError |__InternalError |__ProgrammingError |__NotSupportedError

github_pulls_reviews:

1. exception here is too board. What exception are we expecting? I think we need to be a little bit more specific. And scheduler could always get restarted by runit.
2. @ashb majorly I want to target on the exceptions that come when MySQL connection goes down momentarily, in that case we don't want the scheduler to crash. I should handle this exception specifically

for this issue and for rest of the cases as you said scheduler should shut down. @feng-tao this is the exception:(_mysql_exceptions.OperationalError) (2013, 'Lost connection to MySQL server during query') I should specifically handle this.
3. Does SQLAlchemy give us an exception class that we can catch, so that we don't have to deal with all possible engines available to SQLA?
4. @ashb yes, exception class is : sqlalchemy.exc.OperationalError

jira_issues:

1. **summary:** Scheduler crashes if Mysql Connectivity is lost
description: Airflow scheduler crashes if connectivity to Mysql is lost. Below is the stack Trace Traceback (most recent call last): File "/usr/src/venv/local/lib/python2.7/site-packages/airflow/jobs.py", line 371, in helper pickle_dags) File "/usr/src/venv/local/lib/python2.7/site-packages/airflow/utils/db.py", line 50, in wrapper result = func(*args, **kwargs) File "/usr/src/venv/local/lib/python2.7/site-packages/airflow/jobs.py", line 1762, in process_file dag.sync_to_db() File "/usr/src/venv/local/lib/python2.7/site-packages/airflow/utils/db.py", line 50, in wrapper result = func(*args, **kwargs) File "/usr/src/venv/local/lib/python2.7/site-packages/airflow/models.py", line 3816, in sync_to_db session.commit() File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/orm/session.py", line 943, in commit self.transaction.commit() File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/orm/session.py", line 471, in commit t[1].commit() File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/engine/base.py", line 1643, in commit self._do_commit() File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/engine/base.py", line 1674, in _do_commit self.connection._commit_impl() File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/engine/base.py", line 726, in _commit_impl self._handle_dbapi_exception(e, None, None, None, None) File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/engine/base.py", line 1413, in _handle_dbapi_exception exc_info File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/util/compat.py", line 203, in raise_from_cause reraise(type(exception), exception, tb=exc_tb, cause=cause) File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/engine/base.py", line 724, in _commit_impl self.engine.dialect.do_commit(self.connection) File "/usr/src/venv/local/lib/python2.7/site-packages/sqlalchemy/dialects/mysql/base.py", line 1784, in do_commit dbapi_connection.commit() OperationalError: (_mysql_exceptions.OperationalError) (2013, 'Lost connection to MySQL server during query') (Background on this error at: http://sqlalche.me/e/e3q8) Process DagFileProcessor141318-Process:

jira_issues_comments:

1. raised PR for this: [https://github.com/apache/incubator-airflow/pull/3650]
2. User 'mishikaSingh' has created a pull request for this issue: https://github.com/apache/incubator-airflow/pull/3650
3. User 'mishikaSingh' has created a pull request for this issue: https://github.com/apache/incubator-airflow/pull/3644
4. ashb closed pull request #3650: [AIRFLOW-2703] exceptions from scheduler's heartbeat is handled so that scheduler does not crash URL: https://github.com/apache/incubator-airflow/pull/3650 This is a PR merged from a forked repository. As GitHub hides the original diff on merge, it is displayed below for the sake of provenance: As this is a foreign pull request (from a fork), the diff is supplied below (as it won't show otherwise due to GitHub magic): diff --git a/airflow/jobs.py b/airflow/jobs.py index 04d596b9b7..38f96e2247 100644 --- a/airflow/jobs.py +++ b/airflow/jobs.py @@ -160,33 +160,36 @@ def heartbeat(self): heart rate. If you go over 60 seconds before calling it, it won't sleep at all. """ - with create_session() as session: - job = session.query(BaseJob).filter_by(id=self.id).one() - make_transient(job) - session.commit() + try: + with create_session() as session: + job = session.query(BaseJob).filter_by(id=self.id).one() + make_transient(job) + session.commit() - if job.state == State.SHUTDOWN: - self.kill() + if job.state == State.SHUTDOWN: + self.kill() - # Figure out how long to sleep for - sleep_for = 0 - if job.latest_heartbeat: - sleep_for = max(- 0, - self.heartrate - (- timezone.utcnow() - job.latest_heartbeat).total_seconds()) + # Figure out how long to sleep for + sleep_for = 0 + if job.latest_heartbeat: + sleep_for = max(+ 0, + self.heartrate - (timezone.utcnow() - + job.latest_heartbeat).total_seconds()) - sleep(sleep_for) + sleep(sleep_for) - # Update last heartbeat time - with create_session() as session: - job = session.query(BaseJob).filter(BaseJob.id == self.id).first() - job.latest_heartbeat = timezone.utcnow() - session.merge(job) - session.commit() + # Update last heartbeat time + with create_session() as session: + job = session.query(BaseJob).filter(BaseJob.id == self.id).first() + job.latest_heartbeat = timezone.utcnow() + session.merge(job) + session.commit() + + # Update last

heartbeat time + with create_session() as session: + job = session.query(BaseJob).filter(BaseJob.id == self.id).first() + job.latest_heartbeat = timezone.utcnow() + session.merge(job) + session.commit() - self.heartbeat_callback(session=session) - self.log.debug('[heartbeat]') + self.heartbeat_callback(session=session) + self.log.debug('[heartbeat]') + except OperationalError as e: + self.log.error("Scheduler heartbeat got an exception: %s", str(e)) def run(self): Stats.incr(self.__class__.__name__.lower() + '_start', 1, 1) ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

5. Commit b3918a658c358352348ecb3cb52532e605e7ecf3 in airflow's branch refs/heads/v1-10-test from mishikaSingh [https://gitbox.apache.org/repos/asf?p=airflow.git;h=b3918a6] [AIRFLOW-2703] Catch transient DB exceptions from scheduler's heartbeat it does not crash (#3650) If there is any issue in DB connection then rest of the functions take care of those exceptions but in heartbeat of scheduler, there is no handling for this kind of situation. Airflow Scheduler should not crash if a "transient" DB exception occurs in the heartbeat of scheduler.