Harmony Database Data Path

Cameron Kuchta

October 13, 2018

1 Database Setup

Both the rgt event and rgt check tables should be hardcoded into running. These will not be updated and instead only contain the information about what some event or exit status means.

2 Action Times

Updates to the database should occur whenever a test:

- begins logging
- begins building
- finishes building
- starts running
- finishes running
- starts validating results
- finishes validating

We won't set up a push system but will instead look through the data files for updates and change the database according to those. We can do this for any time step but LSF is cleared once per hour. Thus, in order to get the correct exit status, we need to pull from LSF within that time period.

3 When to Update

Every X seconds, we check the rgt master file. From this we get all of the tests that we may need to look into. We also want to look into all the tests that do not have a done flag on them.

4 Updating

Once we have decided that we want to check on some test, we need to check on all instances of that test that do not yet have a done flag attached to their entry. Since each test instance is labeled by it's harness uid, we can easily cross check this with the database. If the entry does not exist then we need to create such an entry. We wait to do this until we have begun logging the test.

4.1 Beginning to Log

We find if a test has begun logging once the Event 110 file occurs in the test directory. Once this file is created, we parse it into a dictionary. From this we can easily find the necessary information for the `rgt_test` table. Namely,

- Test ID
- Harness start time
- App name
- Test name
- System
- Harness tld (path to where the test is stored)

We also create an event in the `rgt_test_event` table that is associated with this harness uid labeling it as beginning to run.

4.2 Events

When a new event file appears, we create the corresponding event in `rgt_test_event` and change the corresponding status. If it is beginning, we must decide some number for what it means to be currently running. If it is ending, the event status is entered into the table as well.

We always need to check the LSF status of the job once it has begun running. This is because the system may have caused an error that forced the test to interrupt before creating any event files. If it is still running, there is nothing to worry about. If it indeed has been exited for some reason, we record this exit status. We also need to make sure that the next test knows what lsf job to check on while it is building. This is because the next test is built at the end of the current tests run. It is somewhat difficult to correlate the exit status with the an actual problem with the test because it may be due to the test or the test that it is building for the next run or just a system error.

5 State Machine