

Software Engineering Group Project - 7096A

Semester 1 Week 11 - Group 1

Mon May 26, 2008

Ticket 146 - Continuous Testing of Earth Trunk Code

By Filimoni Lutunaika (1154924)

Introduction

In an effort to improve the development and the ultimate quality of the Earth Project application, the developers at Rising Sun Pictures (RSP) have organised an in-house submission process using the Subversion (SVN) source control software. The said process simply involves introducing an automated script that runs the unit tests on submitted source codes before making the commit transaction on the SVN server. Unfortunately, the mere passing of the unit tests alone does not provide adequate assurances on the performance and reliability of the application. To this end, the RSP team had also developed an integration testing script that helps to address this particular shortcoming. Briefly, the test script simulates the performance of the Earth daemon (earthd) in an tightly controlled operating environment with the purpose of identifying faults or breaking points of the application.

At present, the test script *daemon_test.rb* is no longer able to continuously run even on RSP's latest SVN release of the Earth Project (Revision 1180). This can be attributed to recent upgrades of the code to the Rails 2.0 standard and some internal rearrangement of the Earth application source code. A quick scan of the Earth Trac system will show that while the improvements and upgrades have been effected for the core components of the Earth application, the daemon test script remains unchanged. For example, all references to the daemon (earthd) remain as they were before Change-set 1080 (r1080). In the said revision, the script was relocated from *trunk/daemon* to *trunk/test/earthd*.

Task Description

The daemon test script (*daemon_test.rb*) provides a systematic and standardised method of verifying the performance and reliability of any design changes and additions to the Earth application codebase. In its existing state (r1180), there is no other way (apart from the limited scope of the unit and functional tests) to confirm the performance and

robustness of the current state of the Earth application. Therefore, the main outcome of this Ticket is to adapt the daemon test script to the current version of the Earth application codebase.

Implementation

During the investigation, the included paths in the script was appropriately amended in accordance with the current version of the codebase structure. At the same time, the various components of the script were disabled so as to allow for the gradual observance of the various aspects of the script. These disabled aspects were then gradually activated and tested on a case by case basis. A more detailed description of the test script was also documented and is included in the Appendix section.

Results

At the moment, the daemon testing script can be run continuously over the Earth Project codebase and with some minor adjustments (on the View side of the application), the outcome can be conveniently demonstrated through an image-enabled web browser. Admittedly, the script will be continuously amended as and when new and improved features and functionalities are introduced into the Earth Project. However, the extra effort that is required to make the relevant additions or modifications to the script is more than compensated for by the added assurance (upon successfully passing the integration testing phase) that the development of the application is proceeding in the right direction (under the right circumstance).

Conclusion

The daemon testing script was modified to be able to continuously run over the Earth Project codebase. However, further modifications is expected as additional features and functionality of the Earth Project become available in the future.

Appendix A - Revised Daemon Test Script

Source *dtest.rb*

Path *earth_root_dir/test/earthd/*

DaemonTest Class

1. Initialise Rails environment.
2. Checks that the parent directory specified during the execution of the testing script is valid.
3. Set the parent directory as the working directory. This is why the specified directory (refer to above) should be carefully checked as its content will be destroyed and repopulated to test the efficacy of the daemon in updating the varying status of the watched directories. As an added precaution, the working directory is set to a subdirectory of the parent directory called "daemon_test" which is created during execution. However, this added precaution still assumes that there is no existing subdirectory of the specified parent directory named *daemon_test*. If so, then whatever is in that directory is destroyed.
4. Checks for any specified seed value. If none is specified, a random seed is generated based on the current time. This seed value is then displayed to the standard output.
5. This seed value is then used to seed the random number generator through the ruby srand() function.
6. Instantiate DaemonTest object
7. Then call the main_loop method of the DaemonTest object created above. On exiting, display the seed value of the recently terminated daemon test run.

DaemonTest Static Methods

main_loop()

1. First, remove the entry of the parent directory from the earth_test database if it already exists.
2. Set up trap to ensure that script exits when any spawned (child) process terminates.
3. Clear out working directory. Note that this step cannot be tested automatically due to the yet to be implemented remove feature of earthd. However, to enable the execution of the test, a method called clear_parent_directory is called to simulate the clearing of the database. The said method simply removes the current parent directory, recreates the parent directory and waits for 5s to allow the spawned daemon (earthd) to propagate this change to the backend.

4. Repopulate test database using specified parameters that determines the lower and upper bounds/limits to the number of directories and files created.
5. If replay action is specified, then run the `DaemonTest.test_iteration` method for the specified number of times.
6. Launch the daemon (earthd) in the background using the underlying unix fork command.
 - Kill the daemon upon exiting the above background process.
7. Checks that the server has finished indexing the initial directory structure.
8. Verify that the daemon correctly indexes the initial directory structure by calling the `DaemonTest.verify_data` method.
9. Execute iteration of the testing script by calling the `DaemonTest.test_iteration` method.
 - If the number of iteration is not specified, iterate infinitely.
 - Otherwise, only run the iteration for the specified number of times.

verify_data()

1. Test that there's only one directory root.
2. Check that the root directory name (from the database) matches the test root directory.
3. Validate the subdirectories metadata by calling `compare_fs_directory_recursive(root,list)` subroutine.
4. Verify cache integrity by calling the `verify_cache_integrity_recursive(dir)` subroutine.

test_iteration()

1. Check if the replay action is required and performs the relevant mutated iteration up to the specified replayed state.
2. Performs data verification by calling the `verify_data` method at the end of the iteration.

Appendix B - Updated README File

Source *README.dtest*

Path *earth_root_dir/test/earthd/*

===== IMPORTANT NOTICE - READ THIS BEFORE RUNNING THE TEST =====

This document briefly describes the modifications and current status of the dtest.rbscript which is a lean and less comprehensive version of the original daemon_test.rb script.

TESTING NOTES (REQUIREMENTS)

1. Ensure that the test data inserted into the database during the execution of the unit tests are removed. The earth_test database should just be an empty data container but with all the relevant tables and relationship intact.
2. Create a test directory over which the testing script will be working on. Please ensure that you have write access to this directory and that it contains your data. The test script will DESTROY this directory when it is executed.
3. Testing of the caching feature in the daemon is monitored but not acted upon. Further investigation of the caching feature is required before this aspect of the test can be implemented.
4. This minimal version of the daemon_test.rb script was successfully tested on Revision 1180 of the Earth daemon (earthd). It will require some modifications to run on future versions if the tested features have been modified in future version of the Earth daemon.

COMMANDS

1. Testing Over A Specified Number of Iterations

```
./dtest.rb -i nn /path_to_your_test_directory
```

where nn = number of iterations

2. Testing Over A Long Period of Time (Real World Application)

```
./dtest.rb /path_to_your_test_directory
```

3. To Run Testing in Debug Mode

Simply set the flag \$debug_mode to true and re-execute the test script.

Note: The above commands are a subset of the actual commands and testing options that are available on the original test_daemon.rb test script.

SUCCESS CRITERIA

1. The following message should appear on the terminal to indicate successful testing:

```
===== DAEMON TEST PASSED (No Assertion raised) =====
Caching stats: 38/250    [15.2 % ]
=====
```

End