Improving Verification, Validation, and Test of the Linux Kernel: the Linux Stabilization Project

Craig Thomas
Open Source Development Lab
craiger@osdl.org

Abstract

Many studies have been made regarding the development process and configuration management systems of open source software. But relatively little has been done to describe the verification, validation, and testing of open source software.

Small projects tend to have virtually no testing performed by anyone other than the author of the program. Larger projects depend upon a community of users to find and fix problems through use rather than disciplined test methods. The Linux kernel is becoming an exception. In this case an increasing number of professional software test engineers are developing tests and test environments specifically to test the Linux kernel. This paper highlights one project that involves a disciplined test approach to the kernel: the OSDL Linux Stabilization project. The test philosophy behind the blending of commercial software test practices with the methods used in the open source community is presented along with the experiences of creating a collaborative community of testers to contribute to this project.

1. Introduction

It is interesting to note the way project teams form in the open source community. Contributors are attracted to projects that appeal to their interests. In very large projects, the contributors further congregate to their specialized field of expertise. Some projects don't attract any additional contributors other than the creator and thus, the project is maintained by a single author or left to be abandoned if his interest wanes. Sourceforge is littered with abandoned projects.

If a project does not attract a following, the developer of the code is usually left to test the release by himself. Even when the project is a larger collaboration of volunteers, testing is not much better. Much of the testing is performed by simply releasing the code and testing the product through common usage. The model of "release early, release often" is practiced.

The Linux kernel is arguably one of the largest open source projects in existence. While many people contribute features and patches to the kernel, very few people actually take a disciplined approach to testing it. However, within the last 2 years, the approach to testing the kernel has evolved; although the releases are still loaded on various platforms and ad hoc tests are performed by the Linux community, there have been efforts to apply disciplined test methods by contributors interested in verification, validation, and testing.

2. Verification

Linux code patches are typically implemented and submitted to mailing lists for review by the community. Verification takes the form of code reviews by the people interested in that particular part of the kernel. Review comments are posted to the same mail lists and range from constructive criticism and encouragement to blatant rejection of a patch. Often discussions result on the mail list on how to produce the best possible patch. Code reviews are a major component of verification in open source software projects. The more eyes that review code changes, the better.

Sometimes, significant design proposals are introduced to a mail list as well. The design is discussed on a mail list or through IRC channels. The design is verified through an impromptu design review. A recent example is the design and discussion of the sysfs (the Linux kernel device model) feature that redesigned the way devices and busses are represented in the 2.5 kernel.

3. Validation

Typically, patch submitters perform cursory tests of their own code before submitting them to the mail list for review by their peers; however, they rarely submit patches with associated test cases to prove correctness. Exceptions to this behavior include the GNU C compiler project. There, all patches are

submitted with test cases that exercise the code changes.

Since there are no formal specifications for many of the features of the kernel, it is very difficult to validate them. Members would agree on a concept of behavior, possibly relying upon the behavior of a feature in an existing operating system. An email discussion ensues and the feature evolves with a cycle of constant feedback. Rarely are the discussions captured and put into a specification document early on; it would slow down the development of the code. Instead, the specified behavior is left in the mail archives (which may or may not be saved) or later documented after the feature is implemented in the kernel and the behavior is fully understood. Exceptions to this process exist where the kernel features are designed to adhere to a pre-existing document such as POSIX standards or IETF RFCs.

In June 2001, the Linux Standard Base Written Specification 1.0 was released by the Free Software Group to provide a standard for a Linux based application or a Linux run time environment[1]. With the release of the written specification, a beta version of a certification test suite was also produced. As of April 2003, 18 different Linux distributions are listed in their Certification Register as having passed the certification process, which involved passing the certification suite[2]. Although their test suite is not complete, contributions are being made by test developers and new releases are being managed.

4. Testing

A few testers develop micro benchmarks, run the tests against a version of the kernel and repeat the runs with subsequent versions. Results are then posted to the kernel mail lists. The testers are frequently asked to run the tests again with different patches supplied by the requesters. The key to this form of test and check method is that the feedback is rapid; usually a turnaround time for new results is within hours. This is a different approach from commercial operations because there, test cycles are usually planned and scheduled. The test department must complete the other scheduled tests before attempting to rerun a test against a patch. In the Linux community, testers typically know how to apply the developer patch and build the kernel themselves; the feedback of the test result for a new patch is swift. And in this community, rapid feedback is the primary objective of testing. Both approaches greatly benefit from test automation.

Until recently there have not been many efforts to build test suites that provide decent coverage of kernel operations. Over the last couple of years, projects that develop test suites for the Linux kernel have begun to appear on Sourceforge. The Open POSIX Test Suite was created to allow professional testers to develop test cases that validate a correct implementation of the POSIX standards[3]. The Linux Test Project was created to provide a series of functional tests that exercise a large portion of the kernel's system calls[4]. The Scalable Test Platform, developed at OSDL, provides a suite of stress and performance tests to exercise kernel releases[5].

In addition to Sourceforge projects, a new community of professional testers have formed and they are devoting their professional expertise toward testing the kernel. IBM's Linux Technology Center, for example, is composed of many professional testers who have as their primary job responsibility the role of creating test cases for the Linux kernel. They provided the initial work to the Linux Test Project. The center's test suite has received contributions from testers in other organizations as well, showing the growing community of test developers at work.

OSDL is a vendor-neutral non-profit organization dedicated to enabling and guiding Linux development. Within its organization, a group of dedicated test and performance engineers is chartered to create system level tests that can be used to provide reliability and performance data to kernel developers. One significant project recently started is the Linux Stabilization Project. This project has the mission to accelerate the acceptance of the 2.6 Linux kernel by providing test and validation activities to the 2.5 kernel.

5. The Linux Stabilization Project

Kicked off in December 2002, the goal of this project is to provide tests and test results to the Linux community in an effort to promote faster defect fixes and to accelerate the acceptance of Linux 2.6.

OSDL is working with a collaboration of other volunteers to execute tests and collect test results performed at different organizations. All of the test results are collected and retrievable from a single web site. This provides kernel developers with access to these results from a single URL.

5.1. Project Overview

The plan is based upon the execution of correctness tests, stability tests, and performance tests. Correctness tests are basically functional tests that assure the system calls and commands operate as specified. Stability tests incorporate stress and reliability tests to determine how well the kernel sustains loads. Performance tests measure the

performance characteristics based upon a previous, stable release (e.g., version 2.4.20). The tests will be executed on as many different system configurations as possible, thus exposing the kernel to various devices and architectures.

The project utilizes the 2.5 kernel bug tracking database to log defects when found[6]. In addition to the tests outlined above, basic compilation tests will be performed on the kernel source using various configuration files. Errors and warnings are posted to the Linux mail list to identify the problems with the current release.

As part of the OSDL charter, lab equipment will be offered to contributors who want to conduct activities that will aid in the stabilization of Linux. These activities could range from optimizing kernel components to running additional tests.

5.2 Collaboration of Volunteers

In order to gain wide acceptance, various machine architectures must be exercised with a full suite of tests. The systems need to include uniprocessor as well as multiprocessor configurations. It will become necessary to solicit help from others outside this project in order to cover as many architectures as possible. This introduces the difficulties of convincing others of the importance of this effort and to influence them to use some of their time to help with this cause.

The effort started with a small seed of interested volunteers from the Linux Technology Center (LTC) and OSDL. The project started by posting test results from the activities performed from these volunteers, making the results available to the kernel community. Announcements were made to various mail lists telling of the availability of the results. Once credibility for providing useful information has been established, it will be possible to garner interest from other individuals wishing to contribute. A results web page will be available with a simple process for contributing results.

The goal here would be to complement the current work performed by the LTC and OSDL and get the volunteers to test other elements of the kernel, test other architectures, or offer other forms of kernel stabilization such as documentation improvements, performance runs, and scalability tests.

So far, the most difficult part of this effort is getting this information out to the right set of people to let them know this project exists. Posting announcements to the kernel mailing list often get ignored. Success has been found by sending personal communications to individuals that post test results to the kernel mailing list and soliciting their help.

5.3. Tests Utilized

The Linux stabilization effort will utilize several available open source test suites and create new ones as appropriate. Some of the tests available for use are listed below.

- *The Linux Test Project*: a joint project with a good variety of functional tests to validate system calls. http://ltp.sourceforge.net/
- *LSB certification suite*: a functional test suite to show compliance with the Linux Standard Base Specification. http://sourceforge.net/projects/lsb/
- *Open POSIX test suite*: functional test suite to validate POSIX 2001 standards. http://sourceforge.net/projects/posixtest/
- *TAHI Project Test Suites*: a joint project formed with the objective of developing and providing the verification technology for IPv6. http://www.tahi.org/
- *Scalable Test Platform*: a test framework to assist in the testing of a kernel build; it contains a fairly good suite of stress and performance tests. http://www.osdl.org/stp/
- *OSDL Database Test Suite:* A suite of database tests that simulate real-world database workloads. http://www.osdl.org/projects/performance/

In addition to the test suites, this project will utilize OSDL's Patch Lifecycle Manager (PLM). This tool allows a tester to apply a set of patches to a kernel and perform a full kernel build through a simple web interface. By integrating the PLM tool with the Scalable Test Platform, it is possible to automate the process of running a battery of tests against any kernel version released, as well as testing any patch to a kernel. The STP loads the kernel and executes tests on systems ranging from 1 to 8 processors, and from 256 MB to 16 GB.

5.4 Reporting Results

A results page is established at OSDL and offers a list of current stabilization activities[7]. The page serves as a simple anchor point with links to other pages within OSDL or to other sites entirely. A script is created to build the anchor web page every hour so that any new information contributed to the results page is automatically updated. This allows contributions by numerous volunteers and still

provides quick updates. A process is put in place whereby anyone associated with the project can edit the web content. This method has the disadvantage that the content of the web page may have stale or non-existent links if someone inadvertently corrupted its content. But it does provide the advantage that the updates to the page do not have a bottleneck through one person which could prevent rapid updates to new content.

The web content files are saved under CVS so that archives of the files can be easily retrieved in the event of an update failure. The script automatically checks out the latest web file templates and builds a web page.

Test results are analyzed the the volunteers executing the tests and if any defects or anomalies are found, the kernel community is informed through a mail list used by most kernel developers. The most productive postings seem to come from those who provide a comparison to an earlier set of kernel releases and offer an analysis of the discrepancies between test runs. This seems to be highly desired by the kernel developers. The goal of the Linux Stabilization project is to this method and supply comparative information as well as detection of defects.

If a developer has a patch to apply to see if a problem is resolved, the Patch Lifecycle Manager is used to apply the patch to the kernel and rerun the test.

6. Conclusions

The recent involvement of verification, validation, and test by engineers trained in this profession will undoubtedly provide Linux with a needed boost to bringing stability and quality to the kernel. The collaborative efforts of the Linux Stabilization Project will hopefully accelerate the acceptance of the 2.6 kernel by distros. Never before has there been an organized effort to execute a methodical approach to testing a kernel release. The results will provide kernel developers and distros with critical data that will provide them the current state of quality of the kernel.

The Linux Stabilization Project is an attempt to organize a large number of independent test suites and use them to exercise the kernel. It is hoped that this effort will be recognized and appreciated by the kernel community.

Efforts to form collaborations of test volunteers (or any open source development collaborations) are difficult and time consuming. As contributions from this project receive attention from the kernel developers, others will become interested in contributing to this activity and eventually join.

The Linux Stabilization Project is a long term project that is still active. As this project progresses, new information may be attained and additional conclusions may be inferred from that information.

7. References

[1] Free Standards Group Announcement,
http://www.freestandards.org/certification/

[2] LSB Certification Register,
http://www/opengroup.org/lsb/cert/cert_prodlist.tpl

[3] The Open POSIX Test Suite Project,
http://sourceforge.net/projects/posixtest/

[4] The Linux Test Project,
http://ltp.sourceforge.net/

[5] OSDL Scalable Test Platform,
http://osdl.org/stp/

[6] Linux 2.5 Kernel Bug Tracker,
http://bugme.osdl.org/

[7] Linux Stabilization Results Web Page,
http://www.osdl.org/projects/26lnxstblztn/results/