Mobile Integrated Software Testing System

Chris Sosa
Preeyanka Shah

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation
http://www.tdcommons.org/dpubs_series/74

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
Mobile Integrated Software Testing System

Abstract

A mobile integrated software testing system may be a general computing device configured to provide an infrastructure to run automated tests. The mobile integrated software testing system may be a test system loaded with a customized operating system designed to support testing 20-30 devices. A user may able to connect a test device to the mobile integrated software testing system. The user may designate the test that the user wishes to run and the image that the user wants to test against through an interface provided by the mobile integrated software testing system. The mobile integrated software testing system may comprise a server and a scheduler. The user may schedule a test to be run on one of the test devices connected to the mobile integrated software testing system. The mobile integrated software testing system may be configured to display a visual representation of the test results.

Background

During the development of new hardware and software necessary to build a new device such as a laptop, the manufacturer may need to test the software on the new hardware. Due to the delay in time involved in working with a developer of the operating system to test the hardware device, manufacturers may prefer setting up their own lab for testing. This allows manufacturers to quickly address any problems that they may encounter.

Normally, testing new hardware requires many kinds of devices, and setting up the test labs is costly and complex. The complexity of the labs means that it would be difficult for other people to reproduce the testing conditions and reproduce the same error. Furthermore, the software developers have no reliable way to collect results from the manufacturers to improve the operating system and fix any potential software bugs. Even if the software developers of the
operating system are aware of the error, they may find it difficult to reproduce the error and work with manufacturers to fix the error.

For example, to test new hardware, a user may have to obtain a specific version of the operating system that the user wants to test and would have to make sure that the specific version of the operating system is already installed on the new hardware. Installing the operating system on the new hardware may involve figuring out how to get the image for the specific operating system and flashing the new hardware with the image. The user may have to copy the test files to the device to run the tests, and may need to make sure that the versions of the test work correctly with the operating system installed on the device. This process may be very time intensive, and may introduce a lot of errors. Once the test has been performed, the user may need to upload the results to share with the developer of the operating system or other partners, and may need to interpret the results without any visualization.

The subject technology relates to a mobile integrated software testing system with a customized operating system image loaded onto the device. The mobile integrated software testing system is configured to run tests that the software developers of the operating systems normally run to test new hardware and software. These include test for: 1) testing and qualifying firmware; 2) testing the Final Shipping Image (FSI) and Auto-Update (AU) images; 3) adding repeatable qualifications for components to the Approved Vendor List (AVL). This has numerous benefits for the manufacturers and the developers. By allowing tests to be run at the manufacturers’ site, software developers of the operating systems are allowing testing on revisions of hardware that the software developers do not currently have access to. Furthermore, the manufacturers may be able to detect hardware defects before mass production begins and test
new hardware components to replace end of life components or evaluate a new variant of an existing stock keeping unit.

The mobile integrated software testing system may be configured to automatically schedule these tests. Done manually, these tests take human testers multiple days to complete. Even if the tests are not performed in parallel, the tests can be automated, which makes testing more efficient and faster.

**Features of the subject technology**

The idea described herein is a mobile integrated software testing system with a customized operating system that is configured to run tests that the software developers of the operating system normally run to test new hardware. The mobile integrated software testing system may be configured to receive from the user input the test that the user wishes to run. The mobile integrated software testing system may be further configured to receive from the user the software image that the user wants to test against. For example, the user may run a Build Verification Test on the new image. Other tests may include: 1) security testing to ensure that new images retain the security paradigms guaranteed by the hardware device and its hardware; 2) performance testing to collect performance data and ensure that performance regressions do not occur; 3) hardware component verification testing to verify whether new hardware components are eligible to be included in future devices; and 4) autotupdate testing to verify that the newer images are compatible with the operating system update process.

One way to accomplish this may be done through provisioning. Provisioning may allow the mobile integrated software testing system to receive the image that the user wants to test, and compare that image to the image currently installed on the test hardware. If the image that the user wants to test and the image currently installed on the test hardware do not match, then
provisioning will trigger the process of sending over the different bits required to install the image that the user wants to test and instruct the test hardware to re-flash itself with the new operating system. The mobile integrated software testing system requires connection to the test hardware that the user wants to run tests against.

Instead of setting up an expensive lab to test the new hardware, users may be able to test new hardware by plugging the new hardware into a mobile integrated software testing system and selecting the test that the user wishes to run, which may include: 1) testing and qualifying firmware; 2) testing the Final Shipping Image (FSI) and Auto-Update (AU) images; 3) adding repeatable qualifications for components to the ApprovedVendor List (AVL). The mobile integrated software testing system requires virtually no administrative overhead, and storage, network communication, and software updates are all managed automatically.

An exemplary embodiment of the invention is shown below as FIG. 1.
The mobile integrated software testing system 101 may include a Test Framework server and a web application with SSH for remote login. Users may interact with the mobile integrated software testing system 101 on computing device 102 through the web frontend and provided command line tools. Users may run the tests through web frontend or command line utilities provided by the mobile integrated software testing system 101.

Furthermore, the mobile integrated software testing system 101 may establish a test subnet 103 to isolate testing activities and machines from the rest of the corporate network. The test subnet 103 allows users to test various hardware without affecting the corporate network and may be implemented with, for example, a DHCP server. When a new device, such as device 104c, is added to the test subnet, the DHCP client on the new device sends a request to the DHCP server, which assigns an IP address and makes the new device 104c visible by the mobile integrated software testing system 101.

In addition, the mobile integrated software testing system 101 may include a scheduler. The scheduler allows multiple users to schedule tests on a test hardware connected to the same mobile integrated software testing system 101. If there is a plurality of identical test devices, then the scheduler may be configured to schedule the tests on the test device that has the least amount of tests currently scheduled. This enables multiple users to share the test hardware efficiently and test different software without conflicts.

When a test is complete, the user may view the results in the web frontend or retrieve the results from the database for use in other utilities. The results may be displayed as a graphical representation to help the user more easily interpret the results of the tests that were conducted. The results from the test may include test status, system logs, and any failure information.
If the test caused one or more test hardware to return an error, the scheduler in the mobile integrated software testing system 101 is configured to automatically launch a recovery job to attempt to fix the one or more test hardware device. The recovery job may comprise first rebooting the device. If rebooting the device was unable to fix the error, then the recovery job may try to re-permission the device using provisioning. A person skilled in the art would be able to create an algorithm to trigger other types of recovery mechanisms available to the mobile integrated software testing system.

In addition, the scheduler makes it possible to schedule collections of tests periodically or launch tests after a specific event, for example, when a new build completes successfully.

The mobile integrated software testing system 101 may be set up with a private area in a cloud storage solution. The test images and test logs may be uploaded to the private area in the cloud storage. This allows users to easily manage different versions of test images and also to allow users to share reproducible results to other partners so that they may work together to find the root cause of the error.

The software developers of the operating systems may set up an image building and distribution process in order to allow users of the mobile integrated software testing system 101 access to new images to test. Buildbot 107 is a build scheduler that ensures that images are routinely built and uploads the resulting images to a cloud storage system 105. Access to the cloud storage system 105 is granted to the mobile integrated software testing system 101 so that it may retrieve the images for provisioning. The mobile integrated software testing system 101 can also update its own software components via the use of a remote AU Server 106. The mobile integrated software testing system 101 will routinely check the server for an update.
integrated software testing system 101 will download and install the update on a secondary partition and upon reboot, it will attempt to load this new installation.