

Virtual Machine Software Testing

Internship Report

Submitted to:

Student Internship Bureau

University of York, UK



Submitted by:

Mian Asbat Ahmad, PhD. Final Year

Department of Computer Science

University of York, UK

Supervisors: Jennifer Mowat and Ian Broster

Reviewer: Dr. Manuel Oriol

Internship Start Date: 29 April, 2013 and End Date: 19 July, 2013

Preface

This report documents the work done during my summer internship on the project, “Virtual Machine Software Testing” at Rapita Systems Ltd. York, UK. The report includes a brief introduction, a note about the Rapita Systems, project requirements, goals, automating install and uninstall processes, setting the virtual environment, Establishing secure and reliable communication, generating test report, setting front-end for system interaction, reverting modifications after test completion, working of the testing system followed by future work and finally my personal note.

Mian Asbat Ahmad

Acknowledgments

It would not have been possible to achieve the project completion in such a short time without the help and support that I received cheerfully from the whole team of Rapita Systems Ltd. The working environment at Rapita Systems is highly motivational and inspirational. I thank the whole team of Rapita Systems Ltd. in general and Dr. Ian Broster, Dr. Antoine Colin and Mr. Will Lunniss in particular for their valuable suggestions and effective feedback throughout the project period. The whole team generously shared their views and gave useful ideas to work upon. I am also highly indebted to my academic supervisor, Dr. Manuel Oriol for allowing me to avail the unique opportunity and for his valuable guidance and time to review this report.

Last but not the least, I would like to thank the Students Internship Bureau, University of York for creating this wonderful internship opportunity, which enabled me to broaden my knowledge and practical skills.

Mian Asbat Ahmad.

Introduction:

Automation of computer operations can be surprisingly effective and highly beneficial for any organisation. Its initial setup cost may be higher but a quick return on investment outperforms and brings key benefits of cost reduction, productivity, availability, reliability and performance to any organisation. Automation is particularly effective when the nature of job is repetitive and is performed on routine basis. These simple or complex manual jobs consume more time and provide an extra burden on the staff.

Recognizing the need and benefits of automation as a means of staying competitive in the world market, Rapita Systems Ltd. decided to automate the process of software testing. During the internship period of about three months, working for two days per week (eight hours per day), I designed the project and implemented the system developed for automation of the software testing produced by Rapita Systems.

Rapita Systems:

Rapita Systems Ltd. is a local software company based in York Science Park, IT Center, York, UK. It develops software tools for on-target verification, optimisation and code coverage of critical real-time embedded systems. The technology developed by Rapita Systems Ltd is applicable to many different areas of industry and the company has diversified customers including aerospace, automotive and space sectors. Please see <http://www.rapitasystems.com> for more details.

Project Requirements:

I was assigned the task to set up an automated testing environment where the software developed at Rapita Systems would be automatically tested on different operating systems without any user intervention. The project was based on using virtual machines containing various operating systems (Windows and Linux, 32 and 64 bit) and writing scripts to allow software to be automatically installed, executed and tested on different operating systems, with any errors reported correctly. In addition, the project involved working with other software systems of the company, nightly build framework and regression test scripts.

Goals:

The following goals were determined to be the top six.

1. Automating install and uninstall processes of the Rapita Verification Suite software on Windows and Linux operating systems.
2. Setting the virtual environment with the major operating systems.
3. Establishing secure and reliable communication among various systems.
4. Generating detailed test report and its display on webpage.
5. Setting front-end for system interaction.
6. Reverting all the changes after the tests are finished.

The above mentioned goals, each briefly described in the following section, were achieved.

(1) Automating Install and Uninstall Processes:

Some easy-to-use commercial tools are available to automate various tasks but they are costly, unreliable and not flexible enough to fully meet the requirements. We therefore selected tools which were not very friendly and may required more technical skills to handle but were highly reliable, very flexible, open-source and available with no cost.

We selected freely available AutoIT and Expect tools to automate the install and uninstall processes of Rapita Versification Suite (RVS) on Windows and Linux platform respectively. There are 12 steps involved in RVS installation which when performed manually takes around 2 minutes to complete while after automation the process takes not more than 20 seconds.

Automating the install and uninstall processes alone saves a substantial amount of time because of the repetitive nature of job performed after any change in the RVS software. Besides saving time, the automation process also relieves the staff from repeating the same process again and again.

(2) Setting the Virtual Environment:

We selected Oracle Virtual Box to setup the virtual environment, which is a free and open-source software. It is reliable, efficient and highly customisable for meeting the requirements. It provides virtual but very close to the real environment to test software at minimum hardware cost. Snapshots and clones are additional benefits that can be highly effective in restoring the original system after the test completion.

Twelve Linux distributions (32 & 64 bit) and three Microsoft Windows operating systems were installed in Virtual box. These distributions were selected because of their operational at the clients level. The details of each operating system installed in the Birtual-box is stored in the database for later use.

(3) Establishing Secure and Reliable communication:

In the current setup of Rapita Systems, the upgraded version of each software is stored on a dedicated Software Server and the Test Server is used to test the software. The new virtual machine automatic software testing system is configured on the existing Test Server. Moreover, the guest Virtual Machines (VM) are also run on the test server with in virtual environment.

The Software Server, the Test Server and the guest VMs must communicate freely to perform a fully automated testing. For this purpose we created a special user “Rapitabot” in all the three machines and configured it in such a way to establish ssh connection among each other without any key. The same ssh connection is also used to copy the software from Software Server and test scripts from Test Server to guest VMs.

(4) Generating Test Report:

The purpose of the whole testing system is to execute the tests and check that the tests are successful. Therefore, a rigorous reporting system is implemented to track each step of the test execution and log both the pass and fail state.

The logs can be read in the browser by clicking the "Read Log" button or from a log file that is stored in the Test Server. Logs are interactively updated and are readable in real time.

(5) Setting Frond-end for System Interaction:

Special emphasis is given on the front-end to keep it simple, user friendly and accessible from any machine. Therefore, the front-end was developed using PHP: Hypertext Preprocessor (PHP) and HyperText Markup Language (HTML) to allow its access from any machine with browser software.

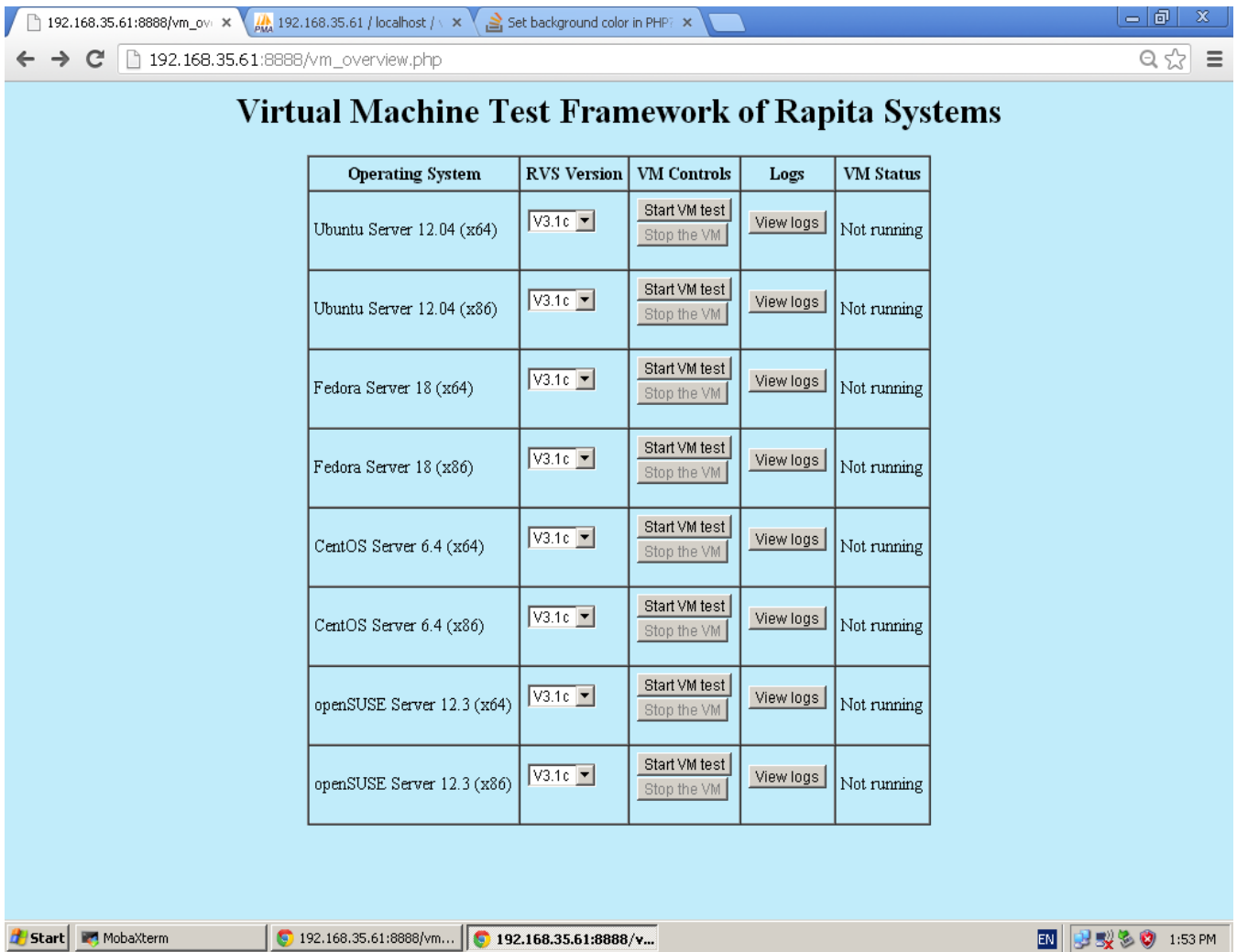


Figure 1: Front-end of Virtual Machine Test Framework of Rapita Systems

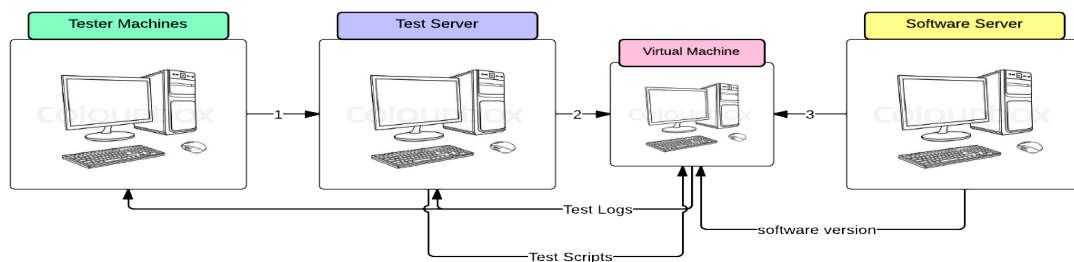
The front-end gets the available software versions from Software Server and the details of Guest VMs from the database to display on webpage. To initiate testing, the tester selects the RVS version with the desired operating system and clicks the “Star VM Test” button. Test is executed in the background and the tester can click the "Read Log" button to view the test results online, which can also be viewed later on from the log file.

(6) Reverting Modifications after Test Completions:

Once the test is completed, it is necessary to delete the modifications in order to avoid any interference in the future tests. A clean up script is placed for the purpose to uninstall the installed RVS software, delete the created folders, script files and safely shutdown the Guest VM.

Working of the Testing System:

The step by step procedure of the system is explained in the Figure 2 with steps given below.



Step 1: Starting Test from Tester Machine

- Step 1.1: The tester browse to the test framework page
- Step 1.2: The tester selects the specific software version from the drop down menu for testing
- Step 1.3: The tester selects the specific Operating System to test the software on
- Step 1.4: The tester click the RUN TEST button and the information is passed to the Tester Server

Step 2: Starting Virtual Machine and Copying Software

- Step 2.1: Test Server analyse the received data and starts the virtual machine accordingly.
- Step 2.2: To get the IP address it performs two steps
 - Step 2.2.1: Find the MAC address of the Virtual Machine
 - Step 2.2.2: Use MAC address to search ARP table for the IP against it.

Step 3: To make sure that system is running and able to accept file

- Step 3.1: Try to establish ssh connection to the system
- Step 3.2: If connection established, then exit and proceed
- Step 3.3: If no connection established, then wait for the system to boot
- Step 3.4: If connection is not established after two minutes then trigger system failure

Step 4: Getting Software and Test Scripts

- Step 4.1: Use scp to copy software from Software Server to Virtual Machine
- Step 4.2: Use scp again to copy the test scripts from Test Server to Virtual Machine

Step 5: Uncompress the software.tar file on Virtual Machine

- Step 5.1: Once the software is copied then use the tar command to untar the software
- Step 5.2: Copy the test scripts from step 4.2 to the new software folder

Step 6: Execute the install test

Step 7: Execute the uninstall test

Step 8: Shutdown the virtual machine

Figure 2: Step by step process of the software testing

Future Work:

Some additional features, beyond the scope of current project were identified which may be explored in the future to make the system more efficient.

These features include:

1. Load balancing on test server to restrict specific number of VMs running at a time.
2. Automatic sending of the test log file as email to the tester after every test execution.
3. Creating more sophisticated test suites to execute after automated installation prior to uninstallation of RVS.
4. Adding more operating systems to the virtual box.
5. Designing more attractive front-end website.
6. Installing new operating system in the Virtualbox through a web page according to the requirements.

Personal Note:

My experience of internship at Rapita Systems was great. The company has a magnificent working environment and great minds contributing to very high quality work. I learnt a lot about test automation, instrumentation, data tracing regression testing and compatibility testing. This internship provided me the opportunity to have professional hands on experience which opened new horizons and broadened my knowledge gained at the academic institution.