

# 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, automation of install and uninstall processes, setting up virtual environment, secure and reliable intercommunication, tracking test execution in test log, design and layout of front-end, reverting changes, working of the testing system followed by future work, conclusion and finally my personal remarks.

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 therefore, thank the whole team Rapita Systems Ltd. special thanks to Dr. Ian Broster, Dr. Antoine Colin and Mr. Will Lunniss for their feedback and valuable suggestions throughout the internship period. They frankly shared their views and gave me useful ideas to work upon. I am also highly indebted to my academic supervisor Dr. Manuel Oriol for allowing me to attain this golden opportunity and for his valuable time to review this report.

Last but not the least, I would like to thank the Students Internship Bureau (SIB), University of York for creating this wonderful internship opportunity, which enabled me to broaden my background knowledge and polish my 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 (QRI) outperforms and brings key benefits of cost reduction, productivity, availability, reliability and performance to any organisation. Automation is particularly effective when the nature of jobs is repetitive and performed on routine basis. These simple or complex manual jobs consume unnecessary 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.

## **Rapita Systems:**

Rapita Systems Ltd. is a local software company based in York Science Park, IT Center, that 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 they have customers in the aerospace, automotive and space sectors. Please see <http://www.rapitasystems.com> for more information.

## **Project Details:**

I was assigned the task to set up an automated testing environment where the software developed at Rapita Systems can be automatically tested on different oper-

ating systems without any user intervention. The project should be set up using virtual machines containing various operating systems (Windows and Linux, 32 and 64 bit). 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 will involve working with other parts of the company's test system, nightly build framework and regression test scripts.

## **Goals:**

The following goals were determined to be the top six.

1. Automation of install and uninstall processes of the Rapita Verification Suite (RVS) software on Windows and Linux operating systems.
2. Setup of Virtual Environment containing all the major Virtual Machines (VM).
3. Establishing a secure and reliable communication between the entities involved.
4. Generating a detailed test report to a file and displayed on webpage on user demand.
5. Setting up of front end with a simple layout to be accessed from any client.
6. Reverting all the changes after the tests are finished.

To achieve the automation of the whole process the project was split up into the following six categories and each one is briefly defined.

## **Automation of Install and Uninstall Processes:**

Although few easy to use commercial tools are available to automate various tasks, however, they are costly, unreliable and not flexible enough to fully meet our requirements. We therefore selected tools that may not be very friendly and may require more technical skills to handle but they are much more reliable, highly flexible, open-source and absolutely free.

We selected freely available AutoIT and Expect tools to automate the install and uninstall process of 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, however, after automation the process takes not more than 20 seconds.

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

## **Setting up Virtual Environment:**

We selected Oracle Virtual Box to setup the virtual environment, which is again a free and open software. It is reliable, efficient and highly customisable to meet our 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 is finished.

Twelve Linux distributions (32 & 64 bit and Desktop & Server) and three Microsoft Windows operating systems were installed in Virtual box. These distributions were selected on the basis of their deployment at Rapita Systems clients. The details of each Guest virtual machine is stored in the database for later use.

## **Secure and Reliable Intercommunication:**

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. Our virtual machine software testing system is configured on the Test server. Additionally the guest Virtual Machines are also ran on the test server with in virtual environment.

These three systems must communicate freely to perform a fully automated testing. For this purpose we created a special user Rapitabot on all the three machines and configured it in such a way that it can establish Secure Shell (SSH) connection between each other without requesting key. The same SSH connection is also used to copy the software from Software server and test scripts from Test server to Guest virtual machine.

## **Tracking test execution in Test Log:**

The purpose of the whole testing system is to execute the tests and check if the tests were successful or not. Therefore, a rigorous reporting system is implemented to track each step of the test execution and log both the pass and fail state.

Test logs can be accessed in two ways. It can be read on 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 can be read in real time.

## **Design and Layout of Frond-End:**

Special emphasis was given on the front-end to keep it simple, user friendly and accessible from any machine. Therefore, we developed the front-end using

PHP: Hypertext Preprocessor (PHP) and HyperText Markup Language (HTML) to allow its access from any machine with browser software. The screenshot of the front-end is given in Figure 1.

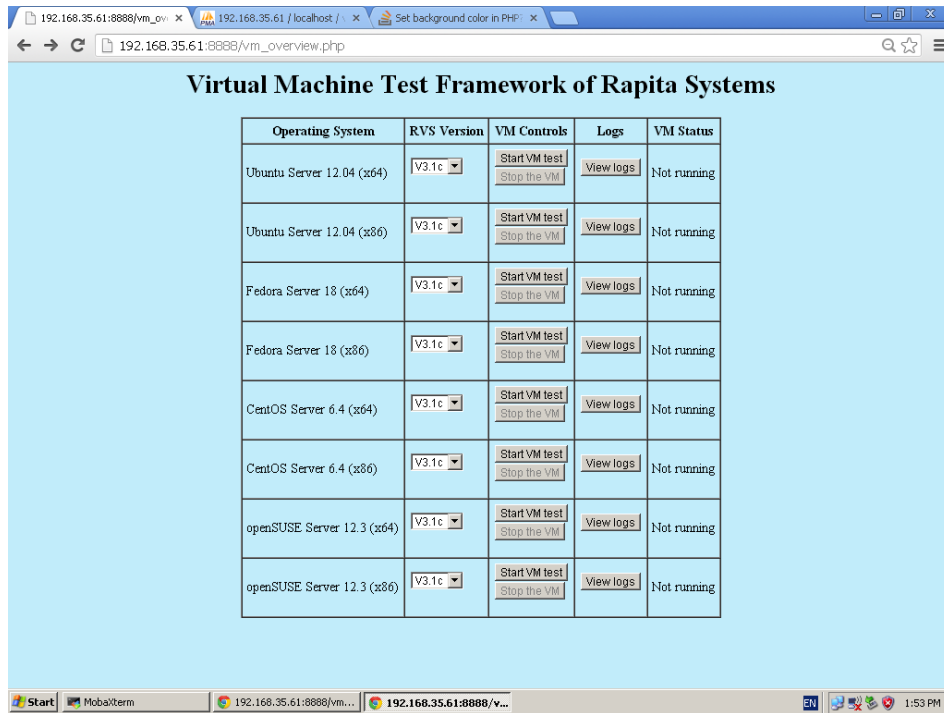


Figure 1: Front-end of Virtual Machine Test Framework of Rapita Systems

The front-end gets the available softwares from Software server and the Guest Virtual machines from the database to display it on a webpage. For testing, the tester picks the RVS version and the operating system and click the test button. Test is executed in the background and user can click the "Read Log" button to view the test results online or tester can check the log file at a later time.

## Revert Changes:

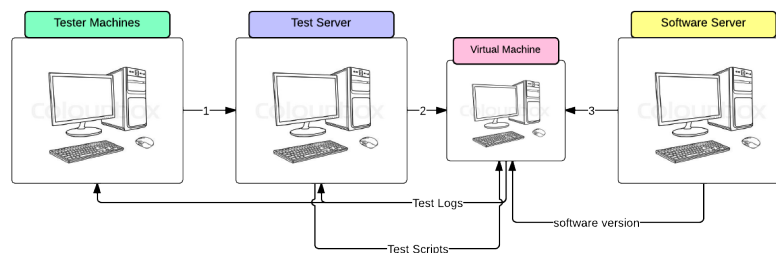
Once the test is completed, it is necessary to delete all its traces to avoid any interference in the future tests. A clean up script is placed for this reason that uninstall the installed RVS software, delete all the folders and script files and safely



shutdown the Guest virtual machine.

## Working of the Testing System:

This section describe the step by step procedure of the system with the help of Figure 2.



### 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, out of the scope of current project, were identified that can be incorporated in the project to make the system more efficient. These features include

1. Load balancing on test server to restrict specific number of VMs running at any time.
2. Automatic Sending the log file as email to the tester after every executed test.
3. Creating more sophisticated test suites to execute after automated installation and before uninstallation of RVS.
4. Addition of more operating systems to the virtual box.
5. Design of more attractive front-end website.
6. New webpage to install a new operating system on the fly in the virtual box according to the requirements.

## **Conclusion:**

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 and regression testing. The project I completed is very satisfactory and highly appreciated by the whole team of Rapita Systems. This internship provided me the opportunity to gain hands on working experience that I couldn't get in the classroom.