CERNVM RELEASE TESTING DEVELOPER MANUAL

CernVM Release Testing - Developer Manual

GNU USER

Technical Report
Version: 0.5 (Draft) June 2011

Abstract

The CERNVM RELEASE TESTING project is a testing infrastructure for CernVM images, the usecase for the project is to provide an automated testing environment, which will install and configure CernVM images, run the set of tests and report the results on a web interface.

Contents

1 Overview

CernVM currently supports images for VirtualBox, VMware, Xen, KVM and Microsoft Hyper-V hypervisors, each new release of a CernVM image needs to be thoroughly tested on each supported platform and hypervisor. The CERNVM RELEASE TESTING project is designed to meet this requirement by providing an automated testing environment for CernVM images, which will install and configure CernVM images, run the set of tests and report the results on a web interface.

The intent of this document is to provided a reference manual on the CERNVM TEST SUITE FRAMEWORK for developers, which should provide enough information about the design that developers should easily be able to add new CERNVM RELEASE TESTING test cases. This developer manual is intended for individuals who have already set up and configured the core components of a RELEASE TESTING infrastructure for CernVM image testing, such as the AMD TAPPER web server and test clients, including hypervisors.If you already have a CERNVM RELEASE TESTING infrastructure set up and wish to further expand and develop the code base, then this guide for you.

All the code needed to begin development of the CERNVM TEST SUITE FRAMEWORK for CernVM image testing is located at the CERNVM RELEASE TESTING Google Code project page[1] including this document and all other documentation.

While this document is not intended to be a replacement for the AMD TAPPER reference manual, the following is a brief description of the RELEASE TESTING infrastructure including an introduction to the core component, AMD TAPPER [2]. Figure 1.1 consists of a diagram outlining the TAPPER Architecture, which consists of test clients and a server, the server is what controls the test clients, gathers results, and then displays the results through a web interface.

The CERNVM TEST SUITE FRAMEWORK was initially intended to only facilitate the role of "Test Suites", which would execute tests and submit a report file in the form of a "Test Anything Protocol" (TAP) file to the "Test Reports Framework", which is essentially the web server that displays the results of tests. But has since been expanded to comprise the role of the "Test Automation Framework", which deploys, installs, and configures the CERNVM images before testing. The most important concept to take away from the diagram is that the CERNVM TEST SUITE FRAMEWORK includes both the "Test Automation Framework" and "Test Suites", even though it is referred to as a "Test Suite".
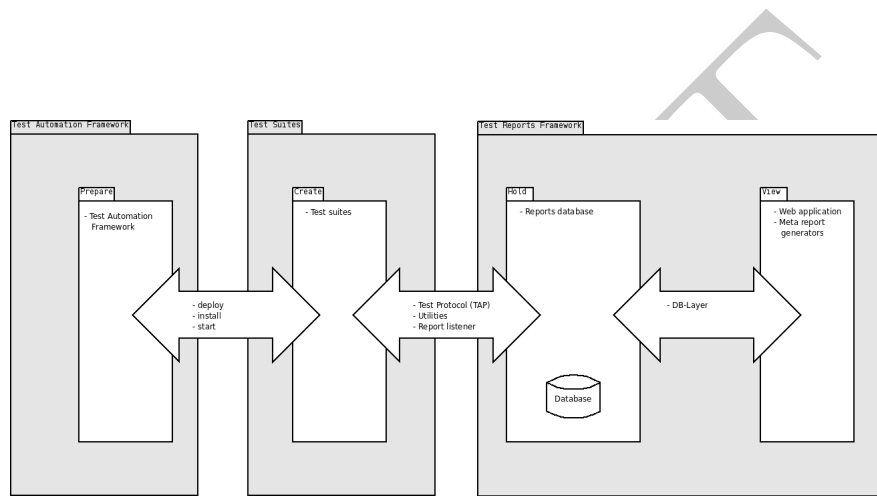
Figure 1.1: Overview of the TAPPER architecture

2 CernVM Test Suite Framework

2.0.1 Framework Design Overview

The CERNVM TEST SUITE FRAMEWORK was initially designed to facilitate the role of "Test Suites" within the RELEASE TESTING infrastructure, which would execute tests and submit a report file in the form of a "Test Anything Protocol" (TAP) file to the "Test Reports Framework". This has since been expanded to compensate for the shortcomings of TAPPER and the CERNVM TEST SUITE FRAMEWORK has since been expanded to comprise the role of the "Test Automation Framework", which deploys, installs, and configures the CERNVM images before testing. This is important to understand as the "Precondition Tests" shown in the following diagrams are mostly tests which facilitate the role of the "Test Automation Framework" by ensuring that the CERNVM image host environment, and the images themselves are properly configured before executing the actual CERNVM RELEASE TESTING test cases.
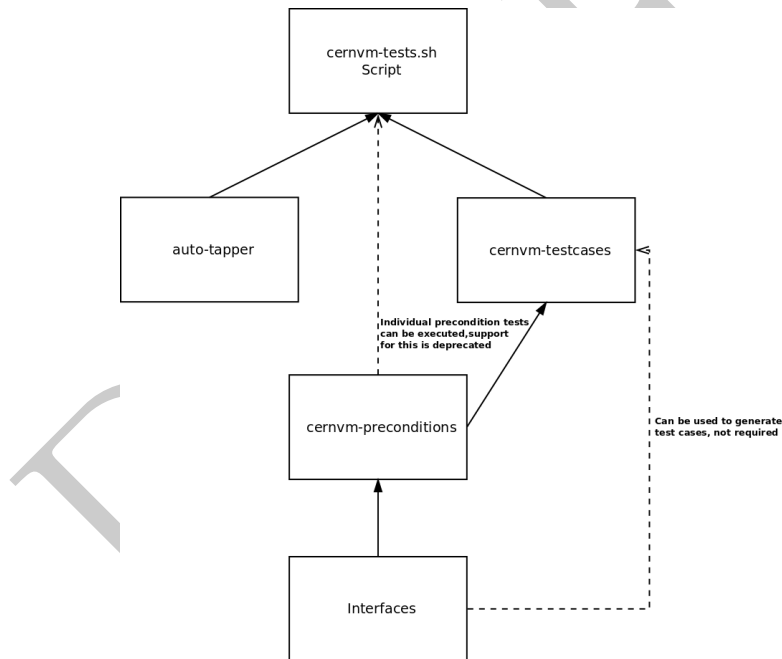
Figure 2.1: Overview of the Proposed CERNVM TEST SUITE FRAMEWORK

3

Currently, due to time constraints the optimal CERNVM TEST SUITE FRAME-WORK design has not been implemented, figure 2.1 is a very simplistic high-level overview of what the *proposed* or intended final architecture is intended to be. The emphasis is on a hierarchical design which is a result in part due to how scope is done in Bash and to limit the functions directly accessed by the **cernvm-tests.sh** script to those provided by auto-tapper and cernvm-testcases. In order for the proposed framework to be implemented, the CERNVM test cases must be modular test cases, independent of each other, this has not been implemented yet and as a result the following diagram outlines the current CERNVM TEST SUITE FRAMEWORK .
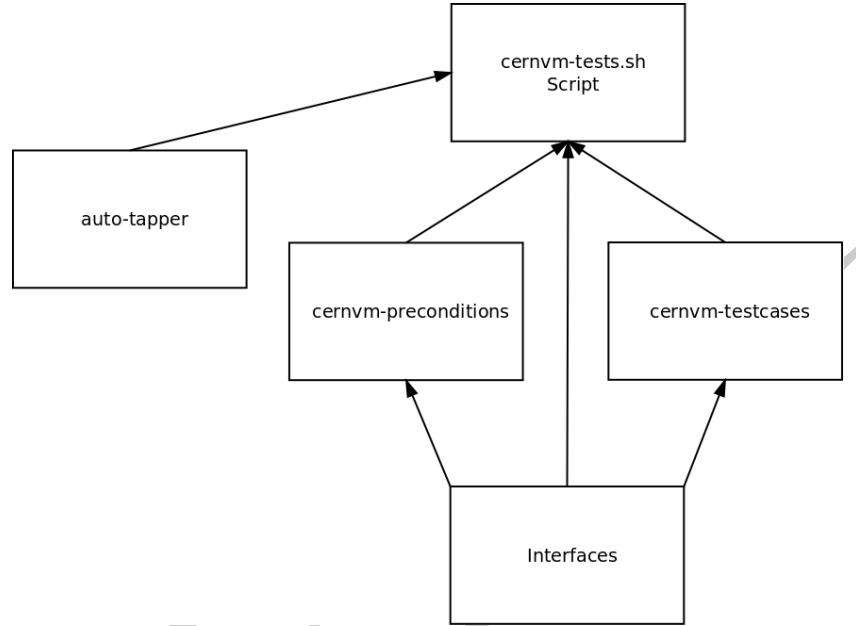

Figure 2.2: Overview of the Current CERNVM TEST SUITE FRAMEWORK

As shown in figure 2.2 the current architecture of the CERNVM TEST SUITE FRAMEWORK differs from the proposed framework because the **cernvm-tests.sh script**, *which is the script that executes the set of* CERNVM *test cases*, requires both the **cernvm-preconditions** and **cernvm-testcases** files. The **cernvm-preconditions** file is what facilitates the "Test Automation Framework" by ensuring that the host environment and CERNVM images are properly configured; the **cernvm-testcases** file is what contains the actual CERNVM RELEASE TESTING test cases, which are required to test the CERNVM image. Inherently, this causes issues as there are precondition tests that must pass before any of the test cases are executed for the results from the test cases to accurate. For example, in order to execute the test case which verifies that the CERNVM image has SSH login support, numerous precondition tests

must first be executed which create and configure the CernVM image and verify that it can be started.
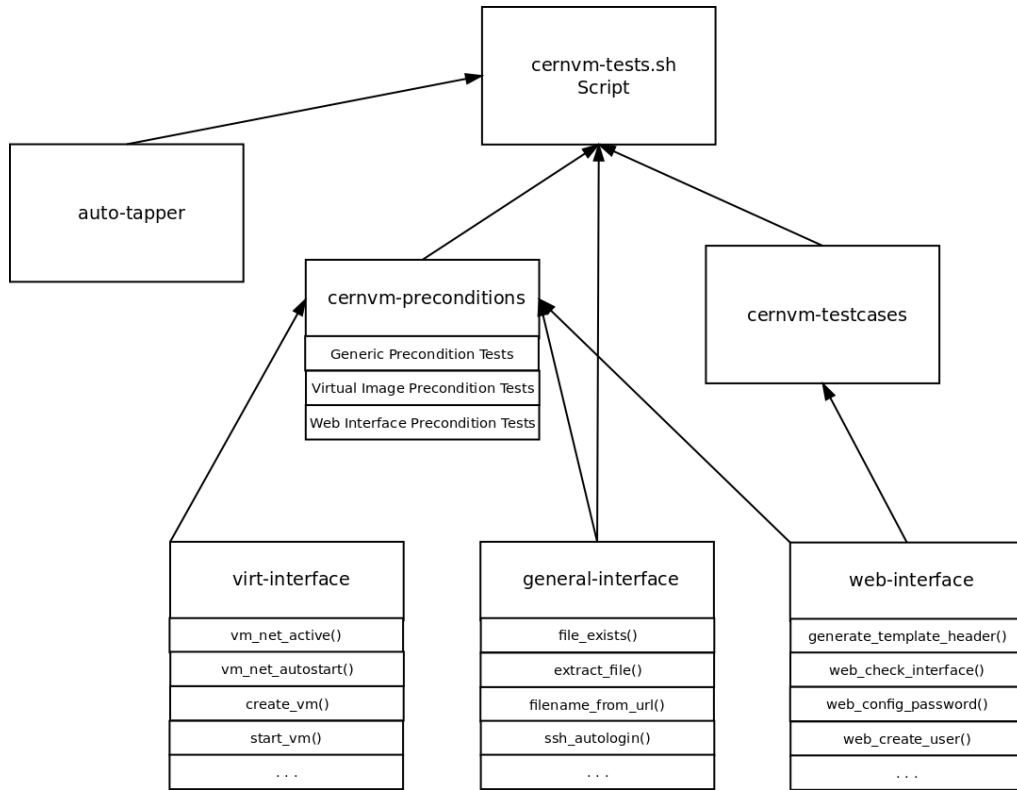

Figure 2.3: Detailed Overview of the Current Cernvm Test Suite Framework

The figure 2.3 provides a much more detailed diagram depicting the relations between the different files which are the individual components that make up the current architecture of the Cernvm Test Suite Framework . As you can see, the hierarchy still exists to an extent but because of the direct dependency the **cernvm-tests.sh** script has on **cernvm-preconditions** and **cernvm-testcases** there are precondition tests which must be executed first and in the correct order before any of the test cases can be executed.

2.0.2 Precondition Tests

Precondition tests are the tests which fulfil the role of the "Test Automation Framework" referred to in the Tapper architecture figure 1.1 within the Cernvm Test Suite Framework . The main purpose of the "Precondition Tests" is to ensure that

the host environment, and the CᴇʀɴVM images themselves are properly configured before executing the actual CᴇʀɴVM Rᴇʟᴇᴀꜱᴇ Tᴇꜱᴛɪɴɢ test cases. The precondition tests configure the host environment and the CᴇʀɴVM images through tests which automate deployment, installation, and configuration of the CᴇʀɴVM images before testing.

Therefore, because the precondition tests are the tests which automate the process of setting up the host environment, the tests must satisfied in order for a CᴇʀɴVM test case to be executed accurately. In a nutshell, they are tests that must be executed, and pass before an actual test case can be executed. Currently, there are three categories of precondition tests,

- Generic Precondition Tests

- Virtual Image Precondition Tests

- Web Interface Precondition Tests

Generic precondition tests, as the name implies, are generic tests which provide functionality to configure the host environment and the CᴇʀɴVM image using methods that are not in the same category as the other two types of precondition. For example, a test that downloads and extracts the CᴇʀɴVM image file, would be an example of a generic precondition test. Unlike generic precondition tests, virtual image precondition tests are very specific to configuring the virtualization environment of the CᴇʀɴVM image and involve tests that interact with the libvirt/virsh library through the **virt-interface** such as creating the virtual machine XML definition file and verifying the that the CᴇʀɴVM image can be started. The last category of precondition tests, web interface precondition tests, are unique in that they are tests directly related to configuring the CᴇʀɴVM image through the web interface of the CᴇʀɴVM image. Although the web interface precondition tests could be expanded to include generic "web" tests, currently the precondition test are limited to configuring and controlling the CᴇʀɴVM image through the web interface.

2.0.3 CernVM Test Cases

- They are the tests which are used to make the cernvm test cases modular and independent of each other (ie. can execute boot errors test case before executing the web restart test case)

3 Test Suite Configuration File

The configuration file is essential to setting up the initial CERNVM test suite for testing, while most of the default settings provided in the configuration file are sufficient for most CERNVM image testing environments, there are still some mandatory settings which **must be configured before testing can begin**. In addition to the mandatory settings that must be specified before tests can be executed, there are also optional configuration settings which provide settings that can override the default settings normally taken when the default configuration file is used, these include options to override the default virtual machine settings specified in the template files.

Each group of settings starts with **CVM**, which is short for CERNVM , but then has a unique prefix depending on the category of setting, there are four categories of options for both the mandatory and optional settings that can be provided. The four setting prefixes are **TS, VM, WEB, TC** these denote options that are specific to a category of configuration options. The following is brief summary of each configuration setting prefix, and what category of configurations each prefix applies to.

TS Options which have this prefix are associated with configuration settings specific to the CERNVM RELEASE TESTING Test Suite

VM Options which have this prefix are associated with configuration settings specific to the CERNVM image hypervisor settings, such as the setting for the virtual machine memory

WEB Options which have this prefix are associated with configuration settings specific to the CERNVM web interface and configuring the virtual machine through the web interface

TC Options which have this prefix are associated with configuration settings specific to the CERNVM Test Cases

3.0.4 Mandatory Settings

In most testing scenarios only the mandatory configuration settings need to be specified such as the hypervisor and the download page, but optional settings are also provided to override internal default settings used by the CERNVM TEST SUITE FRAMEWORK The following is a list of the mandatory settings that must be configured in order for the tests to work, ensure that you enter valid values, *in lower-case*, for the settings indicated.

- CVM_TS_SUITENAME

- Must ALWAYS be set, only define once at the top of the configuration file, usually the default suite name given in the test suite configuration file is fine

- CVM_TS_SUITEVERSION

  - Must ALWAYS be set, only define once at the top of the configuration file, reflects the release version number of the test suite framework, the default suite version given in the test suite configuration should only be changed if you make modifications to the test suite framework which differentiate it from the version released on Google Code.

- CVM_TS_REPORT_SERVER

  - Must ALWAYS be set, only define once at the top of the configuration file, this is the ip address or hostname of the Tapper report server which the reports from the test results are sent to

- CVM_TS_DOWNLOAD_PAGE

  - Must ALWAYS be set, normally the default url provided in the configuration file is accurate, but in the event that the internal CERNVM image release page is relocated then this url must be changed.

- CVM_VM_HYPERVISOR

  - Must ALWAYS be set, MUST be the first setting before the rest of the mandatory and optional settings specific to the hypervisor are set

  - Valid values (case sensitive) are **kvm,vbox,vmware**

- CVM_VM_TEMPLATE

  - Must ALWAYS be set, normally the default template provided in the configuration file should not be changed, only change this to use a custom template file for the CERNVM image

  - The custom template file *must be placed within the templates folder*

- CVM_VM_NET_TEMPLATE

  - Must ALWAYS be set, normally the default network template provided in the configuration file should not be changed, only change this to use a custom network template file for the CERNVM image

  - The custom network template file *must be placed within the templates folder*

  - The network template file, **only applies to kvm and virtualbox**

- CVM_VM_IMAGE_VERSION

  - Must ALWAYS be set, MUST be defined for the HYPERVISOR entry in the configuration file, specifies the version of the CernVM image to use from the release page

- CVM_VM_IMAGE_TYPE
  - Must ALWAYS be set, MUST be defined for the HYPERVISOR entry in the configuration file, specifies the type of CernVM image, such as desktop, basic, head node, etc
  - Valid image types supported, (case sensitive) are **basic and desktop**

- CVM_VM_ARCH
  - Must ALWAYS be set, MUST be defined for the HYPERVISOR entry in the configuration file, specifies the architecture of the CERNVM image
  - Valid architectures (case sensitive) are **x86 and x86_64**

3.0.5 Optional Settings

In most testing scenarios only the mandatory configuration settings need to be specified such as the hypervisor and the download page, but optional settings are also provided to override internal default settings used by the CERNVM TEST SUITE FRAMEWORK. The following is a list of the optional settings that may be specified to override the default settings, the optional settings must be configured for each of the HYPERVISOR settings defined in the configuration file. The optional settings are separated primarily into four categories, host settings, virtual machine settings, web interface settings, and test case settings.

Again, only the mandatory settings are required to be specified in order for the tests to work, the optional settings can be ignored completely and the test suite scripts should still execute correctly. Therefore, optional settings should only be specified by advanced users as improper optional settings can cause precondition tests to return failures, *it is only recommended that you start configuring optional settings after verifying the results of the scripts using only the mandatory settings.*

Optional Host Settings

- CVM_TS_IMAGES_DIR
  - The root directory for the location of the CERNVM images and all configuration files and settings, by default /usr/share/images on Linux/OS X systems and `C:\users\default\application data\images` on Windows systems

- CVM_TS_OSNAME
  - The name of the host operating system, such as Red Hat 5, OS X Snow Leopard, or Windows 7, configure accordingly
  - *Support may be added eventually to automatically configure OSNAME*

- CVM_TS_HOSTNAME
  - The hostname of the system, determined automatically by the script, only set this if you wish to override the default hostname of the system

Optional Virtual Machine Settings

The following are the optional virtual machine settings which can be specified to override the default settings used by the CERNVM TEST SUITE FRAMEWORK these default virtual machine settings used by the framework are based on the virtual machine XML template definition files defined in the templates directory.

- CVM_VM_IMAGE_RELEASE_ID
  - Overrides the default configuration setting, which uses the most recent release id of a CERNVM image, with the specific release id of the CERNVM image to download
  - The release id is used to help better identifying the image, as for each new release added that is the same version, the release id will have incremented to indicate that it is a newer release of the same image version

- CVM_VM_NAME
  - Overrides the default name of the virtual machine set by the virtual machine template XML defintion file
  - It is recommended that this setting is specified if testing multiple versions of the same CERNVM image, for example a name such as "cernvm-vbox-2.4.0" would help differentiate between other versions

- CVM_VM_CPUS
  - Overrides the default number of cpus, which is one cpu by default, set by the virtual machine template XML defintion file
  - Valid values are from **1 - 4**, but the number specified cannot exceed the actual number of cores/cpus on the host system

- CVM_VM_MEMORY
  - Overrides the default default amount of memory set by the virtual machine template XML defintion file
  - It is recommended that you specify this value if thrashing occurs on the CERNVM image when executing tests due to a lack of memory
  - Valid values are in kilobytes and must be based on an amount of memory in kilobytes that is a multiple of a base value of 2. For example, to increase the memory of a system to 1024 MB, set the value as **1048576**, which is the amount of memory in kilobytes

- CVM_VM_VIDEO_MEMORY
  - Overrides the default amount of video memory set by the virtual machine template XML defintion file
  - It is recommended that you specify this value if display errors occurs on the CERNVM image before or when executing tests due to a lack of video memory

- Valid values are in kilobytes and must be based on an amount of video memory in kilobytes that is a multiple of a base value of 2. For example, to increase the video memory of a system to 64 MB, set the value as **65536**, which is the amount of video memory in kilobytes

- CVM_VM_NET_NAME

  - Overrides the default virtual network name set by the virtual machine template XML defintion file

  - This is the one optional setting **you should never configure**, unless you have manually created a different virtual network for the hypervisor

Optional Web Interface Settings

The following are the optional web interface settings which can be specified to override the default settings used by the CernVM Test Suite Framework such as the CernVM image desktop resolution and the primary experiment group.

- CVM_WEB_ADMIN_USERNAME

  - Overrides the default web inteface administration account user name, which is "admin" by default. This optional settings should not have to be modified unless the CernVM web interface defaults change

- CVM_WEB_ADMIN_DEFAULT_PASS

  - Overrides the default web interface administration account password, which is "password" by default. This optional settings should not have to be modified unless the CernVM web interface defaults change

- CVM_WEB_ADMIN_PASS

  - Overrides the web inteface administration account password set by the test suite scripts with a user defined web interface administration password

  - The password specified **must be six characters or longer**

- CVM_WEB_USER_NAME

  - Overrides the default account name "alice" of the new user created by the test suite scripts through the web interface

  - The user name specified should only contain alphabetical characters

- CVM_WEB_USER_PASS

  - Overrides the default password "VM4l1f3" of the new user created by the test suite scripts through the web interface

  - The password specified **must be six characters or longer**

11

- CVM_WEB_USER_GROUP
  - Overrides the default group "alice" for the new user created by the test suite scripts through the web interface
  - The group specified must be a valid group available through the web interface, such as "alice"

- CVM_WEB_ROOT_PASS
  - Overrides the default password "VM4l1f3" of the root account on the CERNVM image set by the test suite scripts through the web interface
  - The password specified **must be six characters or longer**

- CVM_WEB_STARTXONBOOT
  - Overrides the default CERNVM desktop setting set by the test suite scripts through the web interface, which configures X to start on boot
  - The valid values, (lower-case) are either "on" to start X on boot, *which is the default*, or "off" to not start X on boot

- CVM_WEB_RESOLUTION
  - Overrides the default CERNVM desktop resolution, **1024x768** set by the test suite scripts through the web interface
  - The valid values are valid resolutions up to a **maximum resolution of 1680x1050**

- CVM_WEB_KEYBOARD_LOCALE
  - Overrides the default CERNVM desktop keyboard locale, which is "us" by default, set by the test suite scripts through the web interface
  - The valid values are valid locale settings

- CVM_WEB_EXPERIMENT_GROUP
  - Overrides the default CERNVM primary experiment group, which is "ALICE" by default, set by the test suite scripts through the web interface
  - The valid values are one of following group names, **the group name specified must be in UPPERCASE**: ALICE, ATLAS, CMS, LHCB, LCD, NA61, HONE, HEPSOFT, BOSS, GEANT4

Optional Test Case Settings

The following are the optional test case settings which can be specified to override the default settings used by the CERNVM TEST SUITE FRAMEWORK for executing the CERNVM RELEASE TESTING test cases.

- CVM_TC_USER_NAME
  - Overrides the default account name "bob" of the new user created through the web interface as part of a CERNVM RELEASE TESTING test case

- The user name specified should only contain alphabetical characters

- CVM_TC_USER_PASS
    - Overrides the default password "R00tM3" of the new user created through the web interface as part of a CERNVM RELEASE TESTING test case
    - The password specified **must be six characters or longer**

Bibliography

[1] CernVM Release Testing Google Code Project. https://code.google.com/p/cernvm-release-testing/.

[2] Advanced Micro Devices Inc. AMD Tapper. http://developer.amd.com/zones/opensource/amdtapper/pages/default.aspx/, 2011.

14

Index