

Automated Testing with openQA

Alex-P. Natsios
drakevr@2f30.org

Fosscomm 2015 - Athens

7 Nov 2015

\$ whoami

- Alexandros-Panayiotis Natsios
- IRC Handle: Drakevr
- cs undergrad student @ teilar.gr
- openSUSE Advocate

What is openQA?

openQA

- Open Source distribution testing framework
- Can test applications or whole operating systems
- In either GUI or console mode
- was started in 2009
- Is now used by major distributions like openSUSE, SUSE, Fedora

openQA is an integral part of the
development and lifecycle of the
distribution

full test cycle
(pre-validation, validation,
post-validation)

pre-validation

- Incoming changes are “staged” and tested on top of the last good build
- Monitored very regularly
- No submissions are checked in until all openQA tests pass

validation

- In depth validation is done in parallel for every build
- More than 100 validation scenarios tested
- Improved performance and coverage compared to just testing manually

post-validation

- Additional tests can be scheduled automatically after validation passes
- Builds automatically produce verified valid disk images for further testing

Features

- Uses QEMU to fire up Virtual Machine Images
- It can capture images and act on their contents
- It uses libopenCV for fuzzy image matching
- Can generate keystrokes (like a normal user)
- It is mostly written in Perl
- As are the tests
- Rulefiles are written in JSON
- Licensed under GPLv2

New Feature Highlights

- Multi Arch Support (Intel, ppc64le, s390x, aarch64)
- Multi Machine Testing (incl. openvswitch)
- Add On Testing
- Remote Workers
- Real Hardware Testing
- Disk Image Creation
- Testing without Installation
- Dashboard & Comments

Basic Concepts

Jobs

Jobs

One of the most important features of openQA is that it can be used to test several combinations of actions and configurations.

Jobs

For every one of those combinations, the system creates a virtual machine, performs certain steps and returns an overall result.

Jobs

Every one of those executions is called a job. Every job is labeled with a numeric identifier and has several associated settings that will drive its behavior.

Jobs

A job goes through several states:

- **scheduled** Initial state for recently created jobs. (Queued for future execution.)
- **running** Jobs in progress.
- **cancelled** Jobs that have been cancelled by a user or cloned
- **waiting** The job is in interactive mode and waiting for input
- **done** Finished Jobs

Jobs

Jobs in state **done** have typically gone through a whole sequence of steps (called **testmodules**) each one with its own result. But in addition to those partial results, a finished **job** also provides an overall result from the following list.

- **none** For jobs that have not reached the “done” state.
- **passed** No critical check failed during the process.
- **failed** At least one assertion considered to be critical was not satisfied at some point.
- **incomplete** The job is no longer running but no result was provided. Either it was cancelled while running or it crashed.

Exceptions

Sometimes, the reason of a failure is not an error in the tested operating system itself.

- Can be an outdated test
- A problem in the execution of the job
- Other external reason

Recovery

It makes sense to re-run a given job from the beginning once the problem is fixed or the tests have been updated.

Cloning

Every job can be superseded by a clone which is scheduled to run with exactly the same settings as the original job.

Recovery

If the original job is still NOT in done state, then it is immediately cancelled and replaced by the clone.

Recovery

The original job is still retained in the listing as is all the gathered information and results (for future ref and examination since its now considered outdated).

Needles

Needles

One of the main mechanisms for openQA to know the state of the virtual machine is checking the presence of some elements in the machine's 'screen'.

Needles

A needle specifies both the elements to search for and a list of tags used to decide which needles should be used at any moment.

Needles

This is performed using fuzzy image matching between the screen and the so called needles.

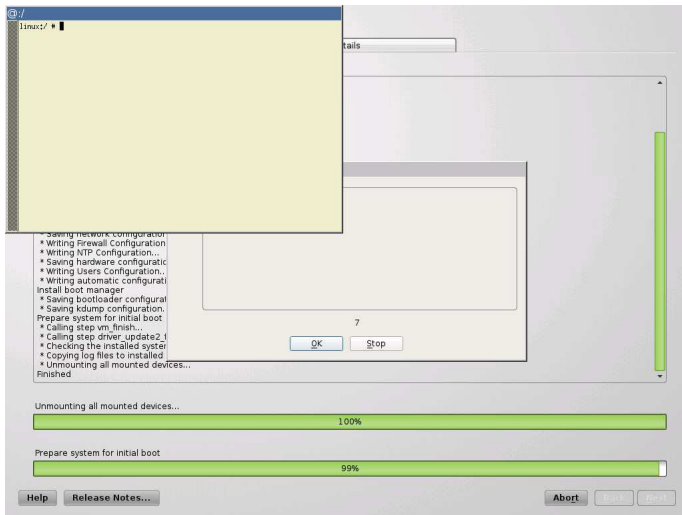
Needle example

A needle consists of a full screenshot in PNG format and a json file with the same name (e.g. foo.png and foo.json) containing the associated data, like which areas inside the full screenshot are relevant or the mentioned list of tags.

Needle example

```
{
  "area" : [
    {
      "xpos" : INTEGER,
      "ypos" : INTEGER,
      "width" : INTEGER,
      "height" : INTEGER,
      "type" : ( "match" | "ocr" | "exclude" ),
      "match" : INTEGER, // 0-100. similarity percentage
    },
    ...
  ],
  "tags" : [
    STRING, ...
  ]
}
```

Xterm Example



Xterm Example

```
{
  "tags": [
    "xterm-in-yast"
  ],
  "area": [
    {
      "xpos": 5,
      "ypos": 22,
      "width": 434,
      "height": 263,
      "type": "match"
    }
  ]
}
```


Interactive Mode

Interactive Mode

There are several points in time during the execution of a job at which openQA tries to match the screen with the available needles, reacting to the result of that check.

Interactive Mode

If the job is running in interactive mode it will stop the execution at that point, freezing the virtual machine and waiting for user input before proceeding.

Interactive Mode

At that moment, the user can modify the existing needles or can create a new one using as a starting point either the current screen of the virtual machine or one of the existing needles.

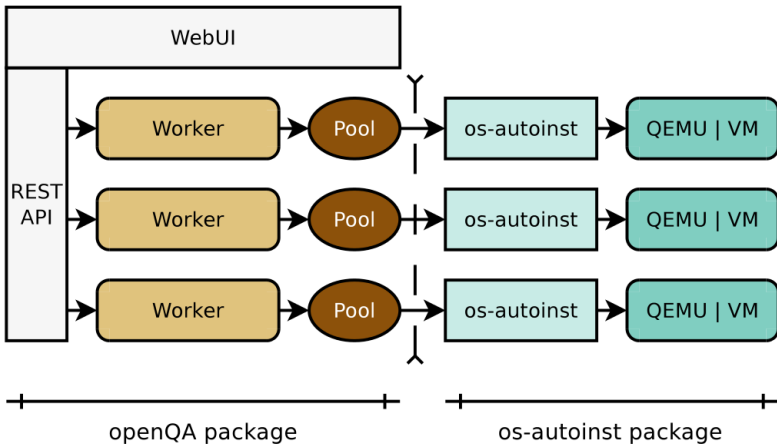
Interactive Mode

Once the needles are adjusted, the user can command the job to reload the list of needles and continue with the execution.

Interactive Mode

The interactive mode is especially useful when creating needles for a new operating system or when the look & feel have changed and several needles need to be adjusted accordingly.

Architecture



Links

- <https://openqa.opensuse.org/> [Homepage]
- <http://en.opensuse.org/openSUSE:OpenQA> [Wiki Portal]
- <https://github.com/os-autoinst/os-autoinst> [Framework Source]
- <https://github.com/os-autoinst/openQA> [Web Interface]

Q & A

Thank you for your attention!

Alex-P. Natsios

`drakevr@2f30.org`

`http://drakevr.gr`

`http://www.linkedin.com/in/drakevr`

`http://www.github.com/drakevr`

`http://www.facebook.com/drakevr`

`http://www.twitter.com/drakevr`