



Overview of openQA

- The heart of openSUSE's automated testing

Max Lin

SUSE Labs



What? openQA?

- It's v3 version now!
- openQA is an automated test tool that can test the whole installation process of an operating system.
- Originally developed by Bernhard M. Wiedemann with contributions from Dominik Heidler.
- Open source! openQA is free software released under the GPLv2 license.
- openQA is implemented in Perl.
- Uses QEMU (by default) for controlling the virtual machines, and uses OpenCV for fuzzy image matching. The backend has integrated with VNC.

Why do we need it?



Condamnés à la migraine sauvez votre tête

Si vous vous contentez de dire, lorsque la migraine vous torture : "C'est un mauvais moment à passer", attendez-vous à passer beaucoup d'autres mauvais moments.

Au lieu d'attendre que l'étau se resserre et que votre tête éclate, faites une cure d'eau de Vichy à Vichy.

Dans la plupart des cas — le vôtre peut-être — la migraine est due à un fonctionnement défectueux de votre foie.

L'eau de Vichy vous rééquilibrera en profondeur.

Un médecin thermal dirigera et surveillera votre cure.

Peu à peu, vos crises s'espaceront, s'estomperont, disparaîtront.

Et pour éviter qu'elles ne reviennent, de retour chez vous, buvez un grand verre d'eau de Vichy avant chaque repas.

Voilà, c'est ainsi qu'il faut agir quand on a les pieds sur terre.

C'est encore le meilleur moyen de sauver sa tête.

faites appel à **vichy** célestins



* <https://flic.kr/p/75utCF> (CC BY-NC-SA 2.0)

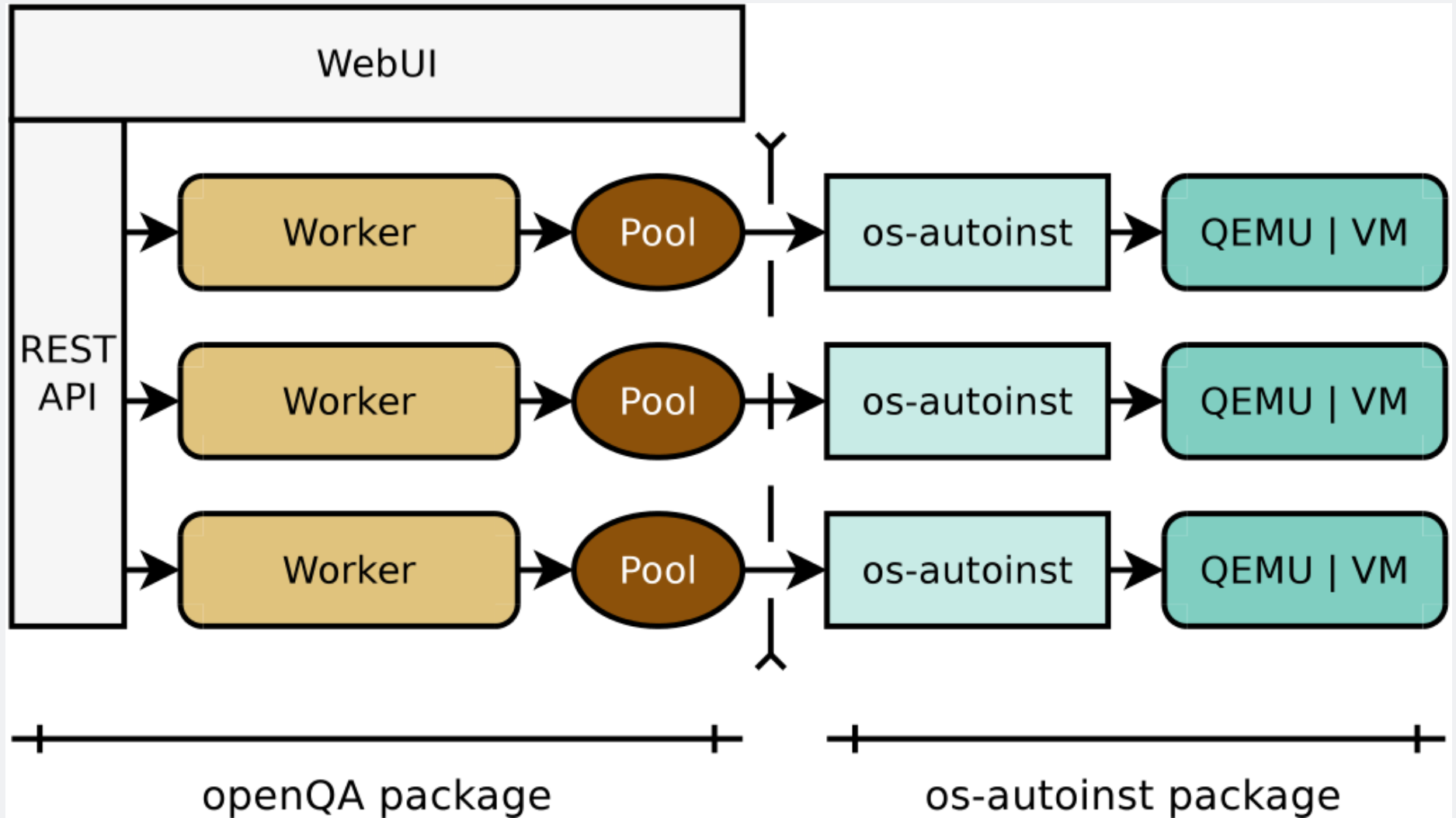
What is it?



What features does it provide?

- Test the whole installation process
- Test distribution update/upgrade
- Regression testing
- Test text/graphic applications
- Test different tests from one image
- Collected the logs once tests failed
- Instant screen output & video recording
- User/access management
- Provides RESTful API
- Provides easy to use webUI

Architecture



Template(Products)

```
{
  "arch" : "x86_64",
  "distri" : "opensuse",
  "variables" : "",
  "settings" : [
    {
      "value" : "1",
      "key" : "DVD"
    },
    {
      "value" : "4700372992",
      "key" : "ISO_MAXSIZE"
    }
  ],
  "version" : "",
  "name" : "oS-DVD-x86_64",
  "flavor" : "DVD"
}
```

Template(Machines)

```
{  
  "settings" : [  
    {  
      "key" : "QEMUCPU",  
      "value" : "qemu64"  
    }  
  ],  
  "backend" : "qemu",  
  "variables" : "",  
  "name" : "64bit"  
}
```


Template(TestSuites)

```
{
  "prio" : 51,
  "name" : "lvm",
  "settings" : [
    {
      "value" : "minimalx",
      "key" : "DESKTOP"
    },
    {
      "key" : "LVM",
      "value" : "1"
    }
  ],
  "variables" : ""
}
```

Template(JobTemplates)

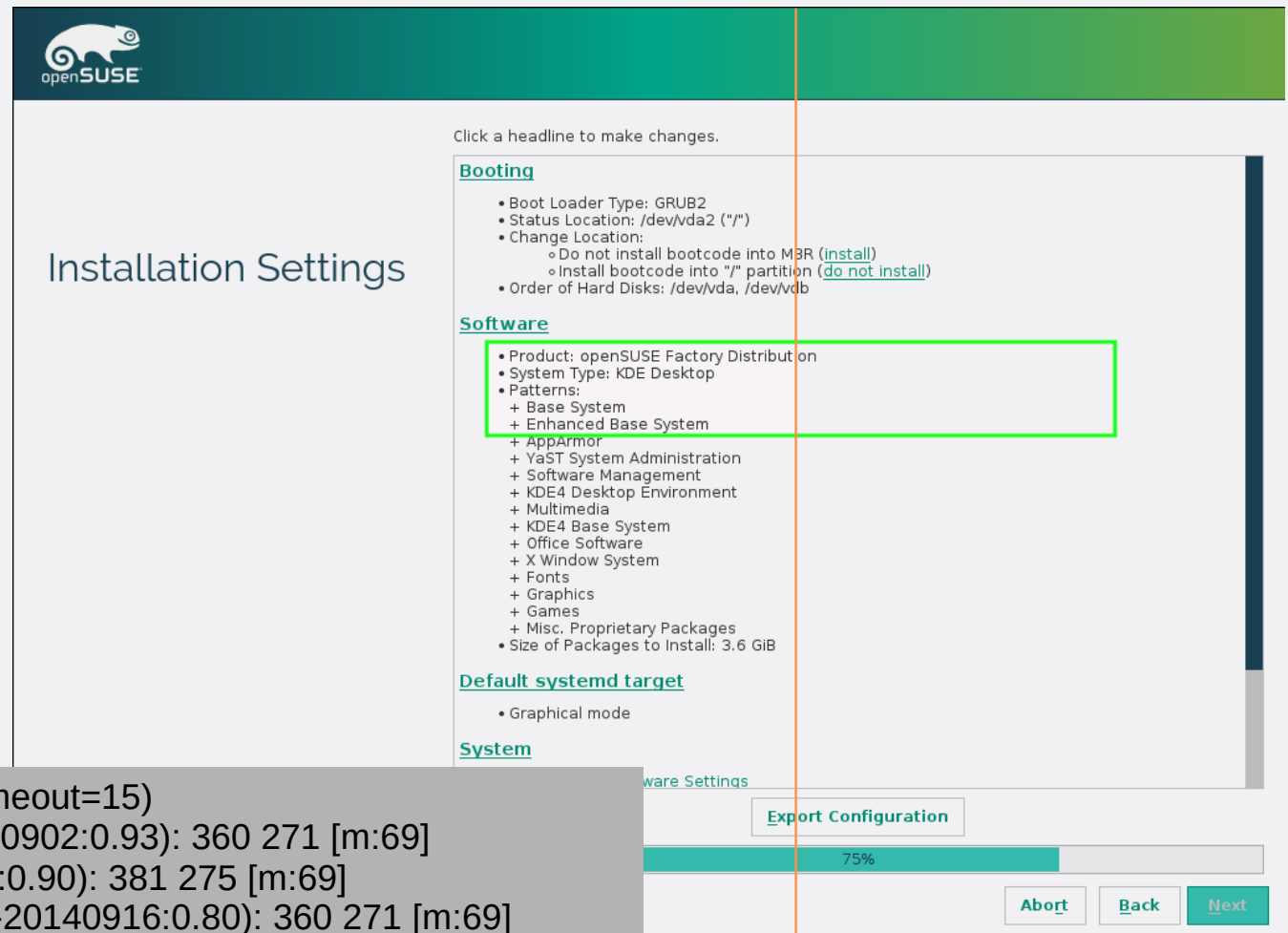
```
{  
  "product" : {  
    "distri" : "opensuse",  
    "arch" : "x86_64",  
    "version" : "",  
    "flavor" : "DVD"  
  },  
  "machine" : {  
    "name" : "64bit"  
  },  
  "test_suite" : {  
    "name" : "lvm"  
  }  
}
```

The needle

- An **screenshot image** with a **json file**

```
{
  "area" : [
    {
      "xpos" : INTEGER,
      "ypos" : INTEGER,
      "width" : INTEGER,
      "height" : INTEGER,
      "type" : ( "match" | "ocr" | "exclude" ),
      "match" : INTEGER, // similarity percentage
    },
    ...
  ],
  "tags" : [
    STRING1,
    STRING2,
    ...
  ]
}
```

Needle matching



```
<<< assert_screen('inst-overview', timeout=15)
MATCH(inst-overview-minimalx-20140902:0.93): 360 271 [m:69]
MATCH(inst-overview-kde-20140912:0.90): 381 275 [m:69]
MATCH(inst-overview-minimalx-13.2-20140916:0.80): 360 271 [m:69]
MATCH(inst-overview-gnome-13.2-20140916:0.82): 362 273 [m:69]
MATCH(inst-overview-kde-20140902:1.00): 381 275 [m:69]
MATCH(inst-overview-gnome-20140902:0.94): 362 273 [m:69]
MATCH(inst-overview-minimalx-20140727:0.21): 399 217 [m:69]
>>> found inst-overview-kde-20140902, similarity 1.00 @ 381/275:
```

Interactive mode&needle editor

Actions

Name:
partitioning-edit-proposal-button-dark-20141003

Tags:
☒ partitioning-edit-proposal-button
☒ ENV-DISTRI-opensuse
☒ ENV-INSTLANG-en_US
 Add

JSON:

```
{
  "tags": [
    "ENV-DISTRI-opensuse",
    "ENV-INSTLANG-en_US",
    "partitioning-edit-proposal-button"
  ],
  "area": [
    {
      "ypos": 402,
      "xpos": 574,
      "type": "match",
      "width": 215,
      "height": 44
    },
    {
      "xpos": 573,
      "ypos": 478,
      "width": 216,
      "height": 44
    }
  ]
}
```

Test modules

boot
☒ isosize
☒ bootloader

inst
☒ welcome
☒ installation_mode
☒ partitioning
☒ partitioning_finish
☒ installer_timezone
☒ logpackages
☒ installer_desktopselection
☒ user_settings
☒ installation_overview
☒ start_install
☒ livecdreboot
☒ first_boot

Screenshot

Needles editor

Source code

Screens.Needle	Image	Areas	Matches	Tags
screenshot				
partitioning-edit-proposal-button-dark-20140902				

Suggested Partitioning

OCR

- Create swap volume /dev/vda1 (1.46 GiB)
- Create root volume /dev/vda2 (8.53 GiB) with btrfs
- Create subvolume boot/grub2/i386-pc on device /dev/vda2
- Create subvolume boot/grub2/x86_64-efi on device /dev/vda2
- Create subvolume home on device /dev/vda2
- Create subvolume opt on device /dev/vda2
- Create subvolume srv on device /dev/vda2
- Create subvolume tmp on device /dev/vda2
- Create subvolume usr/local on device /dev/vda2
- Create subvolume var/crash on device /dev/vda2
- Create subvolume var/lib/mailman on device /dev/vda2
- Create subvolume var/lib/named on device /dev/vda2
- Create subvolume var/lib/pgsql on device /dev/vda2
- Create subvolume var/log on device /dev/vda2
- Create subvolume var/opt on device /dev/vda2
- Create subvolume var/spool on device /dev/vda2
- Create subvolume var/tmp on device /dev/vda2

Edit Proposal Settings

Create Partition Setup...

Expert Partitioner...

Match

Exclude

Help

Release Notes...




















































Abort

Back

Next

13

Tests resultview

Flags	Test	Result	References		
	isotsize	OK			
	bootloader	OK	      		
	welcome	OK	   		
	installation_mode	OK			
	partitioning	OK			
	partitioning_finish	OK			
	installer_timezone	OK			
	logpackages	OK	    		
	installer_desktopselection	OK			
	user_settings	OK	  		
	installation_overview	OK			
	start_install	OK	   		
	livecdreboot	OK	   		
 	first_boot	OK	 		

* Minimal installation with KDE desktop

Tests

- Written in Perl.
- For openSUSE, the tests are collects/defines in **os-autoinst-distri-opensuse**
- Using APIs from **os-autoinst**
- Test case behavior can be controlled via variables
- Milestone? Important? Fatal?
- `is_applicable()` and `post_fail_hook()`

Example

```
sub is_applicable {  
    return $vars{LIVETEST}; # check for $vars{SOMETHING}  
}  
  
sub run {  
    # wait for bootloader to appear  
    assert_screen "bootloader", 30; # timeout 30 seconds  
  
    # press enter to boot right away  
    send_key "ret";  
  
    # wait for the desktop to appear  
    assert_screen "desktop", 300;  
}  
  
sub test_flags {  
    # without anything - rollback to 'lastgood' snapshot if failed  
    # 'fatal' - whole test suite is in danger if this fails  
    # 'milestone' - after this test succeeds, update 'lastgood'  
    # 'important' - if this fails, set the overall state to 'fail'  
    return { important => 1 };  
}
```

**What is it good for
openSUSE?**



Benefits

- An important part of openSUSE development process
- Make sure new installation not broken
- Main packages won't breaks current Factory
- Did the testing work before manual testing
- Developer/Packager could know the issue at the first moment

Where to get it?

- GitHub <https://github.com/os-autoinst>
 - **openQA** containing documentation, server, worker and other support scripts.
 - **os-autoinst** with the standalone test tool and the API for tests.
 - **os-autoinst-distri-opensuse** containing the tests
 - **os-autoinst-needles-opensuse** with the needles associated to the tests
- Binaries(RPM)
 - http://software.opensuse.org/package/openQA?search_term=openQA

Join&Contact

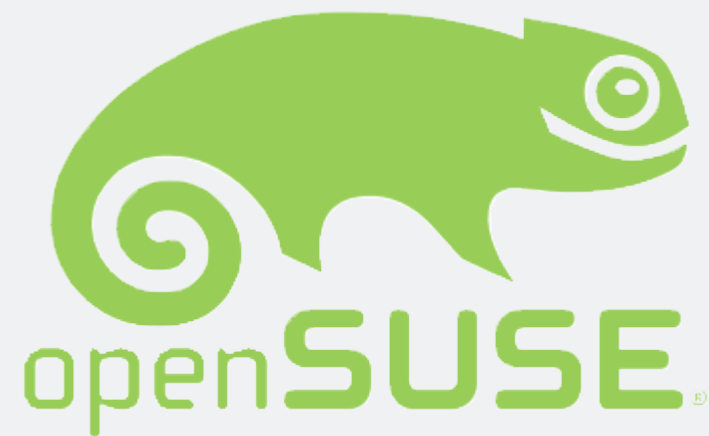
- openSUSE project management tool
 - <https://progress.opensuse.org/projects/openqav3>
- #openSUSE-factory IRC channel at Freenode
- opensuse-factory@opensuse.org

Questions?



Thank you!





Join us on:
www.opensuse.org





Overview of openQA

- The heart of openSUSE's automated testing

Max Lin

SUSE Labs



What is it?



**What is it good for
openSUSE?**



Questions?



Thank you!



