1 Problem Framing

2 Resolution strategy
The juggler dilemma

It is *hard* to juggle

- Code/new feature development
The juggler dilemma

It is *hard* to juggle

- Code/new feature development
- Bug handling
The juggler dilemma

It is *hard* to juggle

- Code/new feature development
- Bug handling
- Write tests for all the features
How can we accommodate testing?

Test tools *have* to be easy

- To understand
How can we accommodate testing?

Test tools *have* to be easy
- To understand
- To hack
How can we accommodate testing?

Test tools *have* to be easy

- To understand
- To hack
- To do useful things with
QA vs developer level testing

If the testing tools (written for QA) are hard to understand:

- “Ok, I’ll write my own tool”.
QA vs developer level testing

If the testing tools (written for QA) are hard to understand:

- “Ok, I’ll write my own tool”.
- It is fine. Then you start repeating things over and over.
KVM autotest: The hard path

KVM autotest was written with focus on QA level testing

• It has grown to cover libvirt and other virt backends
KVM autotest: The hard path

KVM autotest was written with focus on QA level testing

- It has grown to cover libvirt and other virt backends
- It can do migration, virtio console, hotplug, among others

Lucas Meneghel Rodrigues  lmr@redhat.com

KVM autotest: It is not just a QA tool anymore
KVM autotest: The hard path

KVM autotest was written with focus on QA level testing

- It has grown to cover libvirt and other virt backends
- It can do migration, virtio console, hotplug, among others
- But has a steep learning curve
How to solve the problem

Layered approach

- Keep the code base working for the original cases

KVM autotest: It is not just a QA tool anymore
How to solve the problem

Layered approach

• Keep the code base working for the original cases
• Get rid of useless things for the development use case and expose the essentials
How to solve the problem

Layered approach
- Keep the code base working for the original cases
- Get rid of useless things for the development use case and expose the essentials
- “I don’t care about autotest, just give me a test suite”
Split test modules from autotest core

Going after the layered approach

- Autotest will provide a stable test API
Split test modules from autotest core

Going after the layered approach

- Autotest will provide a stable test API
- Test modules are developed independently
Split test modules from autotest core

Going after the layered approach

- Autotest will provide a stable test API
- Test modules are developed independently
- Merged all virt types in a single test module
Turn virt tests into a separate suite

Virt tests still have all the code to work under autotest

- It works as a separate testsuite, with autotest deps
Turn virt tests into a separate suite

Virt tests still have all the code to work under autotest

- It works as a separate testsuite, with autotest deps
- After a bootstrap stage, just execute a simple runner
Minimize deps: JeOS

Nowadays this is another buzzword: “Just enough OS”

• In real life, people need a small guest to run tests on
Minimize deps: JeOS

Nowadays this is another buzzword: “Just enough OS”

- In real life, people need a small guest to run tests on
- Very small Fedora 17 x86_64 sparse image
Minimize deps: JeOS

Nowadays this is another buzzword: “Just enough OS”

- In real life, people need a small guest to run tests on
- Very small Fedora 17 x86_64 sparse image
- Easy to maintain, fairly small compared to a full DVD

KVM autotest: It is not just a QA tool anymore
Test runner

Present the tests in a minimalistic way

- Terse, unittest like output

Lucas Meneghel Rodrigues  lmr@redhat.com  

KVM autotest: It is not just a QA tool anymore
Test runner

Present the tests in a minimalistic way

- Terse, unittest like output
- Can list available tests
Test runner

Present the tests in a minimalistic way

- Terse, unittest like output
- Can list available tests
- Provide a qemu path and a test list and you’re good
Demo
Changes in autotest (last year)

Lots of work under the hood

- Namespace fixes and cleanups
Changes in autotest (last year)

Lots of work under the hood

- Namespace fixes and cleanups
- Improved release management
Changes in autotest (last year)

Lots of work under the hood

- Namespace fixes and cleanups
- Improved release management
- Fedora packaging work done
Outline
Problem Framing
Resolution strategy

Changes in autotest (last year)

Lots of work under the hood

- Namespace fixes and cleanups
- Improved release management
- Fedora packaging work done
- Stand alone RPC client

Lucas Meneghel Rodrigues 1mr@redhat.com

KVM autotest: It is not just a QA tool anymore
Roadmap

What now?

- Allow to run tests written on any language

KVM autotest: It is not just a QA tool anymore
Roadmap

What now?

- Allow to run tests written on any language
- Evolve core functionality into libraries
Roadmap

What now?

- Allow to run tests written on any language
- Evolve core functionality into libraries
- Run tests out of tree (say, qemu tree)
Roadmap

What now?

- Allow to run tests written on any language
- Evolve core functionality into libraries
- Run tests out of tree (say, qemu tree)
- You all are welcome to help
Contact

- cleber@redhat.com
- lmr@redhat.com
- Virt test devel list (virt-test-devel@redhat.com)