KVM autotest: It is not just a QA tool anymore

Lucas Meneghel Rodrigues
lmr@redhat.com

KVM Forum 2012
November 5th, 2012

The juggler dillema

It is *hard* to juggle

- Code/new feature development

The juggler dillema

It is *hard* to juggle

- Code/new feature development
- Bug handling

The juggler dillema

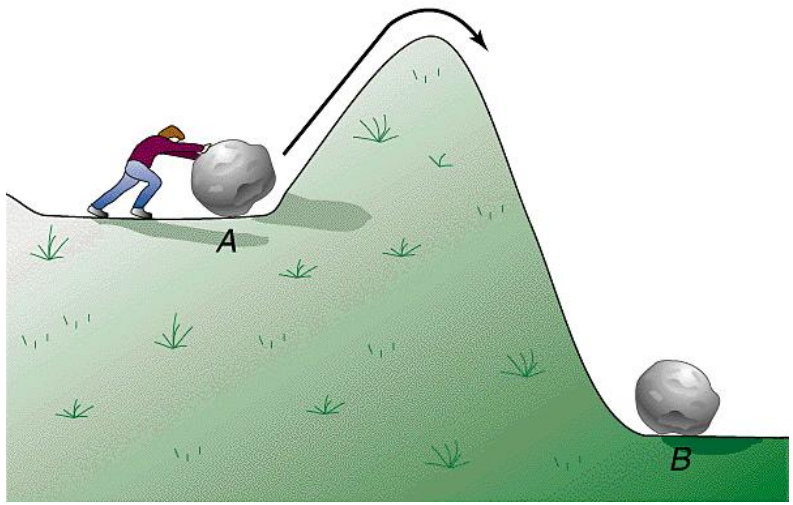
It is *hard* to juggle

- Code/new feature development
- Bug handling
- Write tests for all the features

How can we accomodate testing?

Test tools *have* to be easy

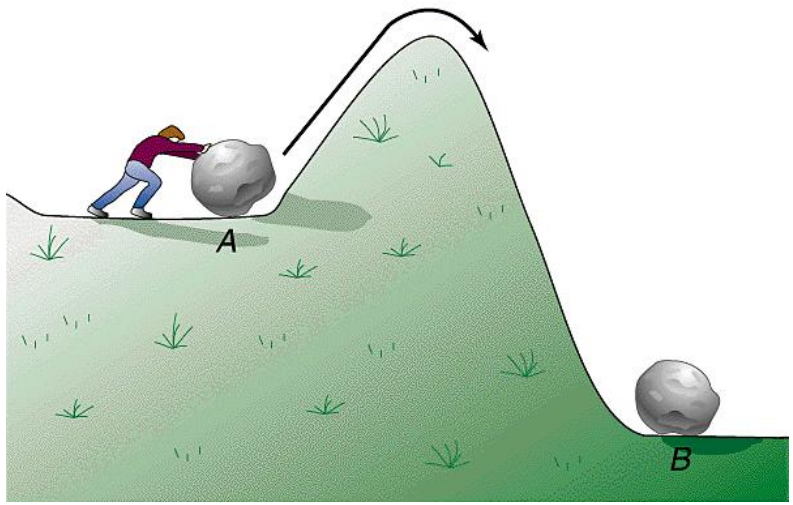
- To understand



How can we accomodate testing?

Test tools *have* to be easy

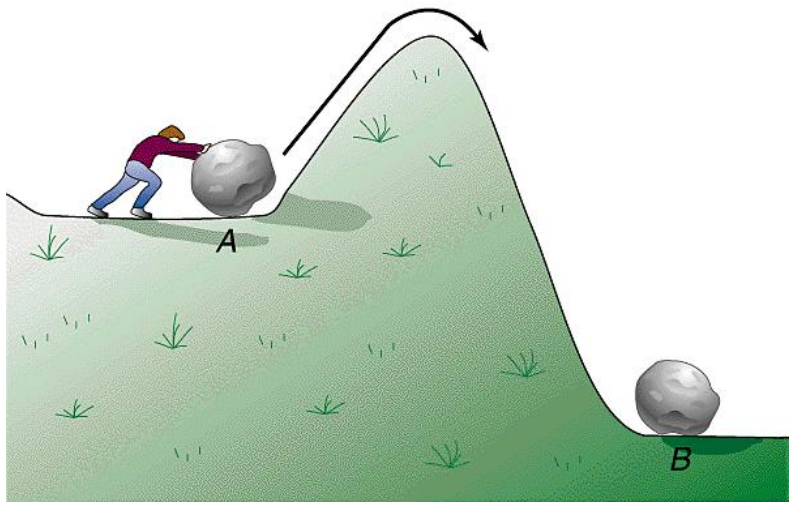
- To understand
- To hack



How can we accomodate 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

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

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

Test runner

Present the tests in a minimalistic way

- Terse, unittest like output

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

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

Roadmap

What now?

- Allow to run tests written on any language

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*

Questions?

Contact

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