Autotest Introduction

Amos Kong
akong@redhat.com

Jan 7, 2011

Agenda

- Autotest framework intro
- How to use Autotest
- How to add new tests to Autotest

Resources

- http://autotest.kernel.org/
- http://www.linux-kvm.org/page/KVM-Autotest
- Upstream maillist: autotest@test.kernel.org
- RH internal IRC channel: #autotest

Autotest framework intro

- A framework for fully automated testing
- Designed primarily to test the Linux kernel
- Useful for many other functions such as qualifying new hardware
- An open-source project under the GPL (2006)
- Used and developed by Google, IBM, Red Hat and others
- 2010-05-24 - Autotest 0.12.0 released

http://test.kernel.org/tko/

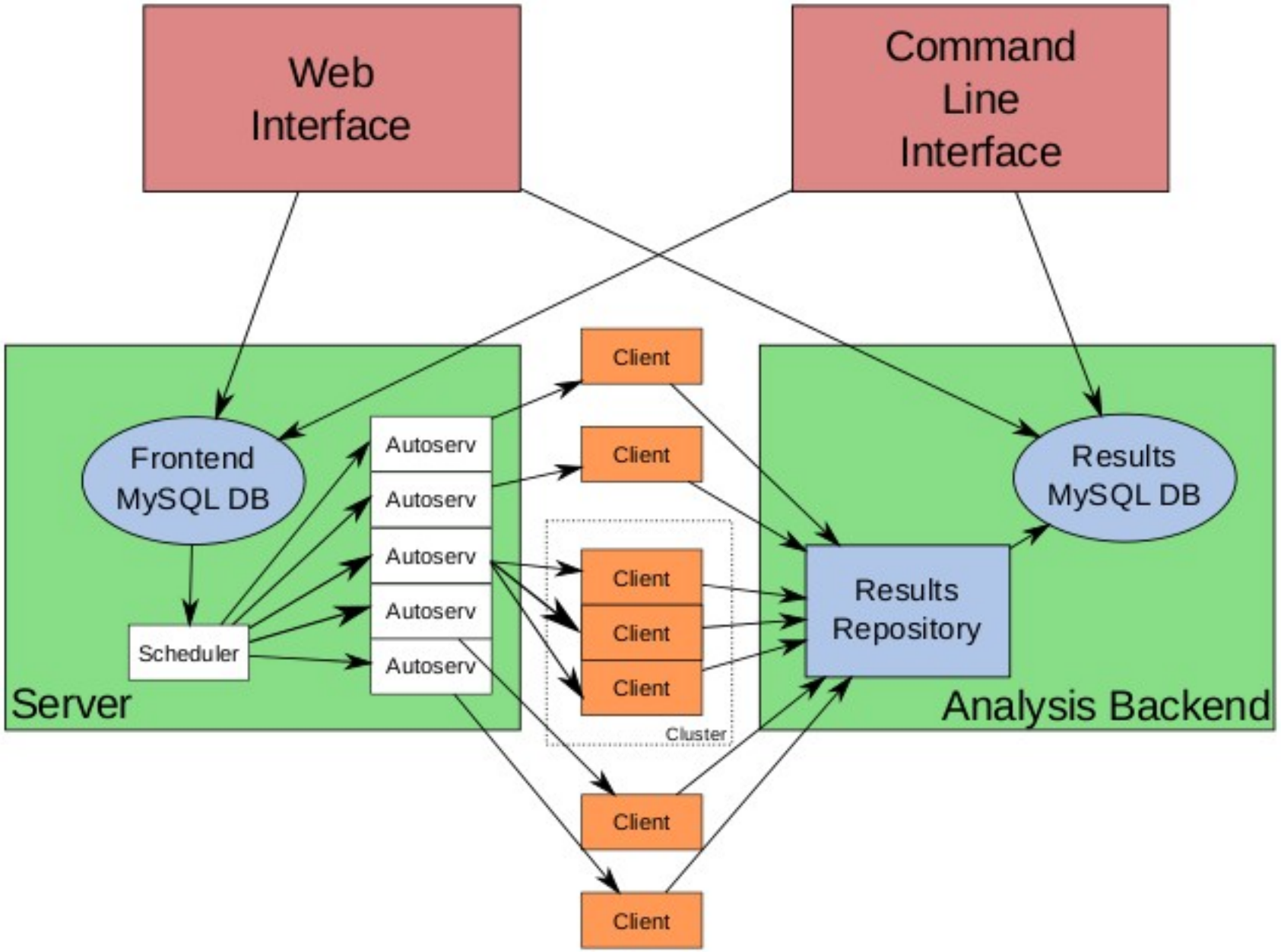
Functional **Performance** **Crackerjack** **[About Page]**

Column: machine_group ▼ Row: kernel ▼ Condition: Help Submit

Name your query: Save Query View saved queries

0 % | to 75 % | to 85 % | to 90 % | to 95 % | to 100 %

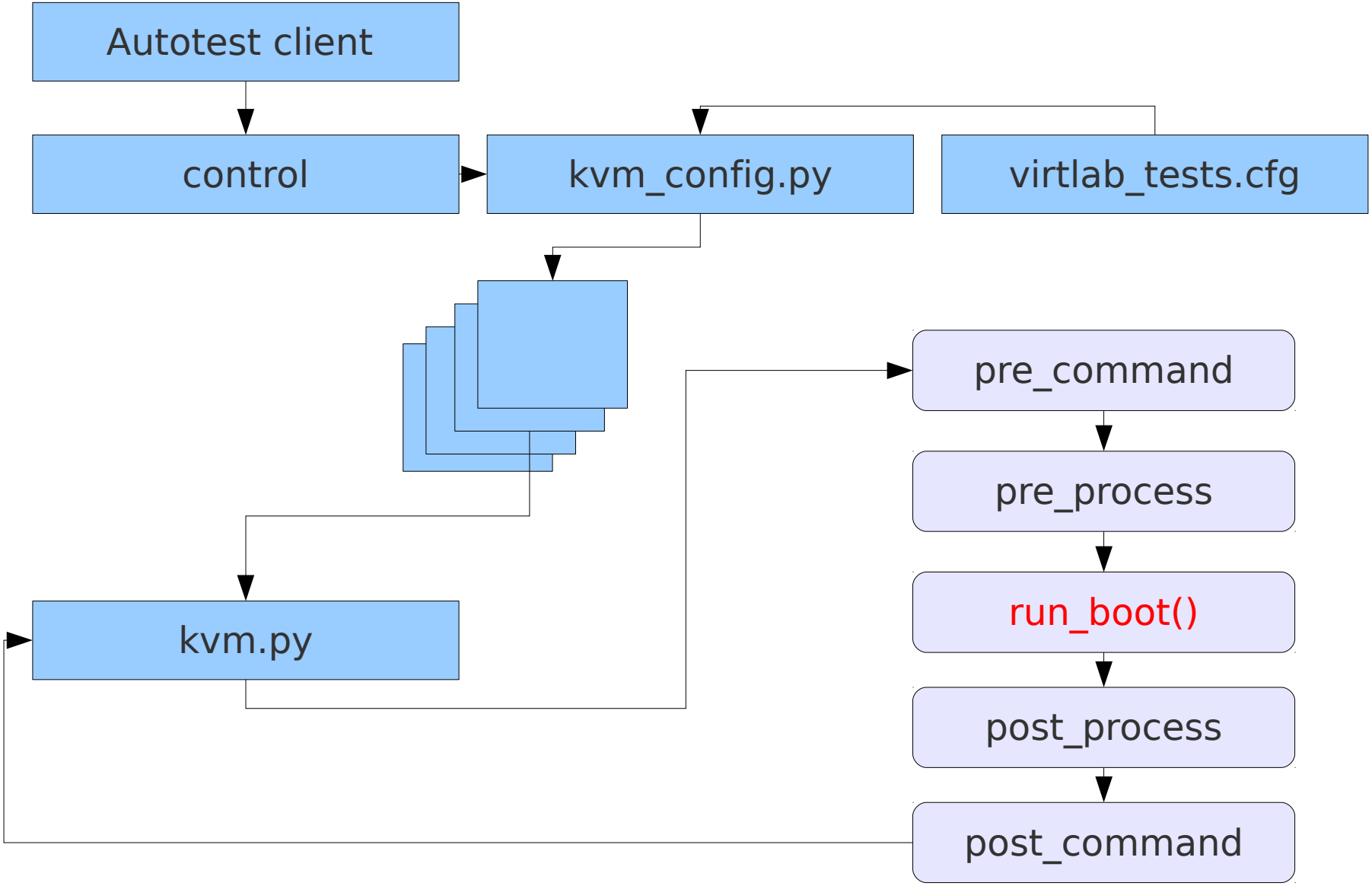
(Flip Axis)	i386	powerpc	s390	x86-64
UNKNOWN				7701 / 8152
2.6.36.1				100 / 100
2.6.36				160 / 172
2.6.35.9				100 / 100
2.6.35.8				300 / 300
2.6.35.7				299 / 300
2.6.35.6				300 / 300
2.6.35.5				300 / 300
2.6.35.4				300 / 300
2.6.35.3				300 / 300
2.6.35.2				300 / 300
2.6.35.1				300 / 300
2.6.35-git16				100 / 100


Web Interface
Command Line Interface
Frontend MySQL DB
Autoserv
Autoserv
Autoserv
Autoserv
Autoserv
Scheduler
Server
Client
Client
Client
Client
Client
Cluster
Client
Client
Results MySQL DB
Results Repository
Analysis Backend

Autotest (http://autotest.kernel.org/) is a set of libraries and programs used to automate regression and performance tests on the linux platform. Composed by:

- Client: Engine that executes tests in test machines
- Server: Copies client code to the test machines, triggers test execution, monitors machine/test status and brings back test results to the server machine
- Scheduler: Schedules test jobs according to user input, creating server processes for each job, and stores results on autotest's test database
- Frontends: Allows users to run jobs and visualize test results conveniently

KVM-Autotest client



KVM-Autotest: APIs and features

- Define large test matrices
- Reuse processes between tests: An environment file
- An expect-like library to control qemu processes (kvm_monitor.py)
- Build and install KVM from several methods (release tarballs, git, brew/koji, rpms)
- Fully automated install of several kinds of Linux and WinXP–Win7
- Serial output collection and login / SSH connection log/ Screendump
- Capture and analysis core dump on qemu segmentation faults
- Run the latest qemu-kvm unittests
- Ways to install Virtio-win drivers and run WHQL testing

- Good power management support (fence)
- Host system info collection

Main files inside the KVM test folder

- kvm.py: KVM test main entry point. It is a simple loader of the subtests
- kvm_config.py: Parser of the configuration file format
- kvm_preprocessing.py: Functions to modify the environment
- kvm_subprocess.py: Expect like library
- kvm_utils.py and kvm_test_utils.py: Utility functions
- kvm vm.py: The modeling of a KVM virtual machine. Implements its methods by spawning kvm subprocess instances of qemu

Anatomy of a KVM-Autotest subtest

A KVM-Autotest test implementation boils down to implementing a python function using the test API to accomplish what you need to do, which is usually something along the lines:

- Get a living VM from the test environment
- Establish remote sessions to the Vms
- Send commands to the remote sessions on the VMs, verify their return codes, capture their outputs
- Send commands to the qemu monitor, verify their return codes, capture their outputs
- Determine whether the test has passed or failed based on this info

How to use autotest

- Git repo:
  git://github.com/autotest/autotest.git
- Easy guides:
  http://www.linux-kvm.org/page/KVM-Autotest/Client_Install
  http://www.linux-kvm.org/page/KVM-Autotest/Server_Install

Autotest Server Installation

- Install required packages
- Important server configuration for the KVM test
- Setup MySQL
- Install extra packages
- Update Apache config
- Update Autotest config files
- Run DB migrations to set up DB schemas and initial data
- Adding some initial frontend objects
- SELinux Issues
- Restart Apache
- Test the server frontend
- Start the scheduler
- Update kvm test config files

Autotest Client Installation

- Install required packages and get Autotest
- For the impatient

  # autotest/client/tests/kvm/get_started.py

1. Verifying directories
2. Creating config files from samples
3. Verifying iso
4. Verifying winutils.iso
5. Checking if qemu is installed
6. Checking for the KVM module
7. Verify needed packages to get started

- Run the test: Install Fedora13 guest, boot, shutdown

  # client/bin/autotest client/tests/kvm/control

How to add new tests to Autotest

- Create a directory with your test name in autotest/client/tests/

Ex: tsc/

- Add a control file and a python file with test name in tsc/, Ex: control tsc.py

  class tsc(test.test): setup() /initialize() /run_once()

- Run your test, and keep developing until you're satisfied

  # client/bin/autotest client/tests/tsc/control

How to add new tests to KVM-Autotest

- Create a python file with your test name inside the subfolder tests. Ex: boot.py
- Implement a function run_boot() in boot.py
- Add test parameters to tests base.cfg.sample

```
- boot:
    type=boot
```

- Modify one of the test sets under tests.cfg to include your test
- Run your test, and keep developing until you're satisfied

```
# client/bin/autotest client/tests/kvm/control
```

Demo

- Register a machine to Autotest Server
- Submit jobs in Autotest server
- Execute Autotest subtest: tsc
- Execute KVM-Autotest subtest: boot

Question?