

# **Avocado: Open Source Testing Made Easy**

## **LinuxCon North America, 2015**



**August, 17th, 2015**

Lucas Meneghel Rodrigues <lmr@redhat.com>  
Senior Software Engineer

# Agenda

- What is Avocado?
- Architecture
- Features
- Demo
- Roadmap

# Who we are

- Virtualization Test Team @ Red Hat
- We develop testing tools for KVM and Libvirt
- We maintain Autotest and virt-test
- The experience with those prompted us to imagine what the next generation of testing tools would look like

# Without further ado...

- *Avocado is a set of tools and libraries to perform automated testing on linux platforms*
- Developed to reconcile the needs of different teams involved in software development: *QE* and *Development*

# Testing tools: *QE* vs. *Development*



***Let's crash and burn it!***  
if we can't crash it, we're not  
working hard enough



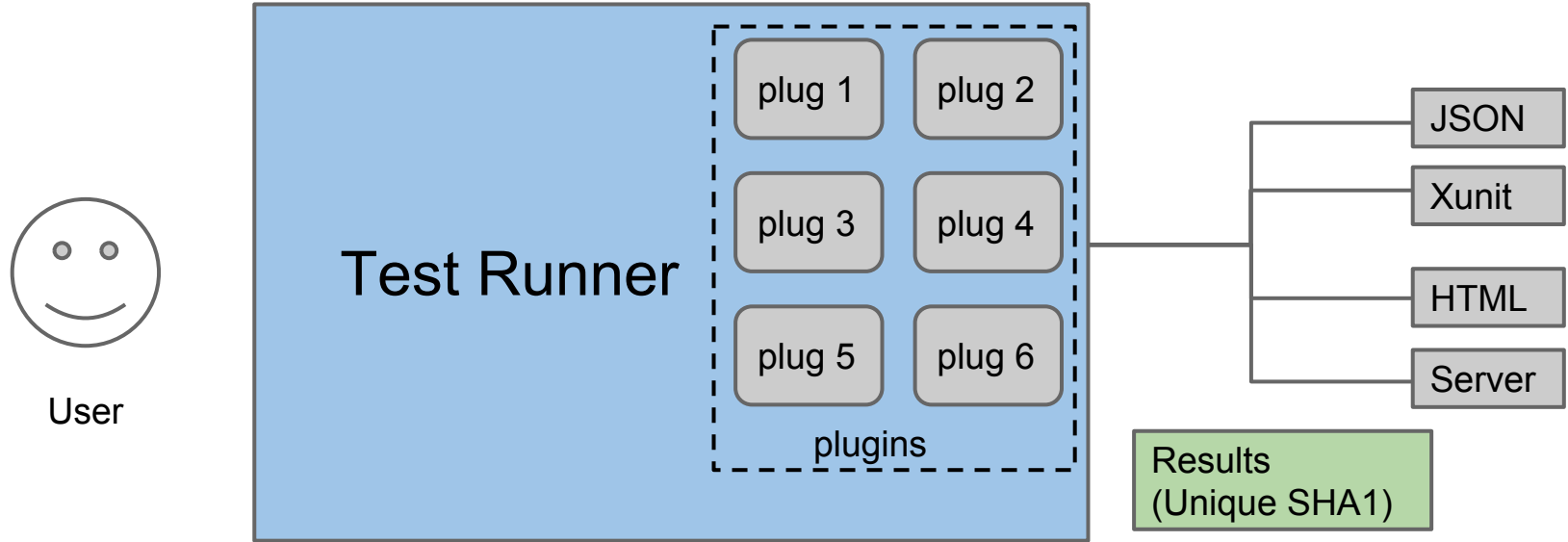
***In code we trust (Works For Me)***  
we write good code – and keep  
getting better at it

# Avocado: A new testing toolbox

*Testing should be fun and simple:*

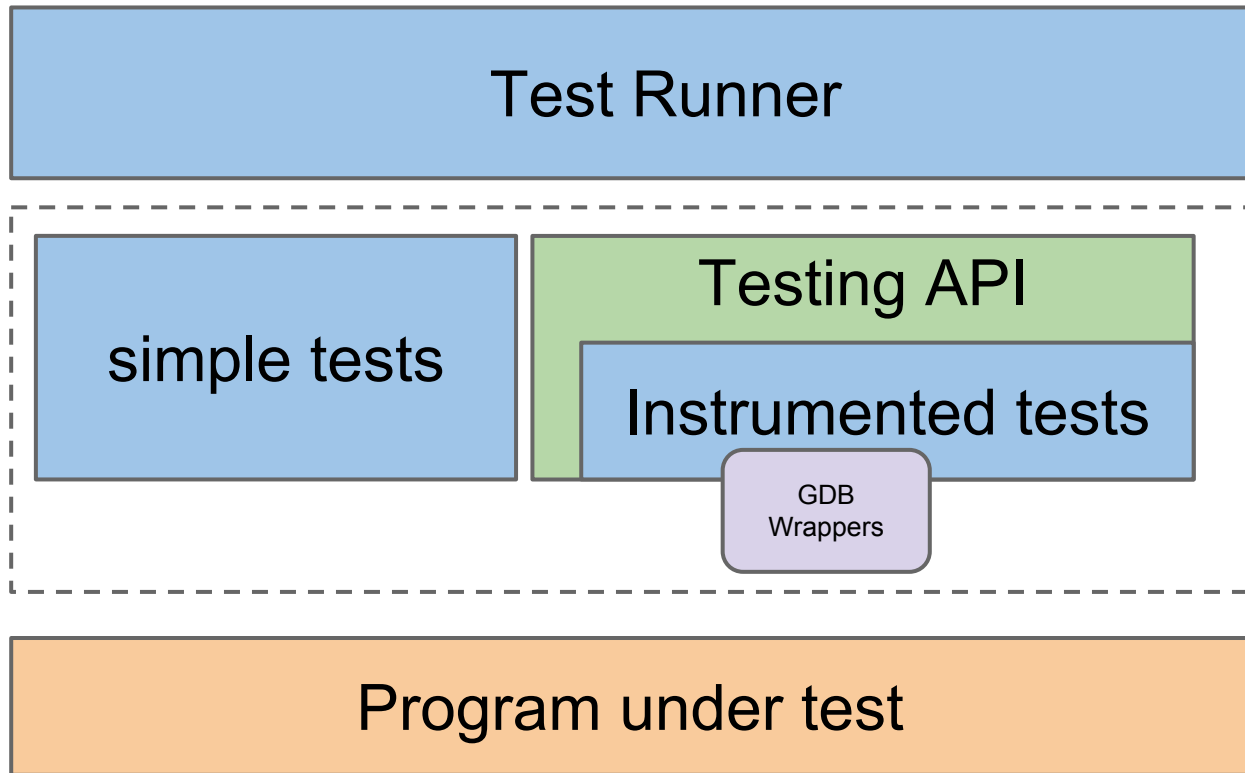
- Start with a test runner, with optional features helpful for debugging and development
- Add more building blocks (plugins) as you need more features
- Don't restrict test development choices - Use any language you want (you get benefits from using test APIs though)
- The same test runner is used in the infrastructure that runs CI jobs - the *test grid*

# Avocado For Users



# Avocado For Test Writers

Test Writer





# Multiplexer

- In virt testing, we have large test matrixes
  - Disk formats
  - NICs
  - Guest OS
  - Host OS
- The multiplexer is a mechanism of describing a test matrix in a compact way
- YAML based
- Allows the use of filters to reduce the scope of the matrix

# Multiplexer - simple example

bread: !mux

italian:

bread: Italian

nine\_grain\_wheat:

bread: Nine grain wheat

topping: !mux

american:

topping: American

monterey\_cheddar:

topping: Monterey cheddar

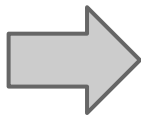
filling: !mux

roast\_beef:

filling: Roast Beef

tuna:

filling: Tuna



Variants generated:

Variant 1: /bread/italian,/topping/american,/filling/roast\_beef

Variant 2: /bread/italian,/topping/american,/filling/tuna

Variant 3: /bread/italian,/topping/monterey\_cheddar,/filling/roast\_beef

Variant 4: /bread/italian,/topping/monterey\_cheddar,/filling/tuna

Variant 5: /bread/nine\_grain\_wheat,/topping/american,/filling/roast\_beef

Variant 6: /bread/nine\_grain\_wheat,/topping/american,/filling/tuna

Variant 7: /bread/nine\_grain\_wheat,/topping/monterey\_cheddar,  
/filling/roast\_beef

Variant 8: /bread/nine\_grain\_wheat,/topping/monterey\_cheddar,/filling/tuna

# Multiplexer - complex example

```

env: !mux
  production:
    malloc_perturb: no
    gcc_flags: -O3
  debug:
    malloc_perturb: yes
    gcc_flags: -g

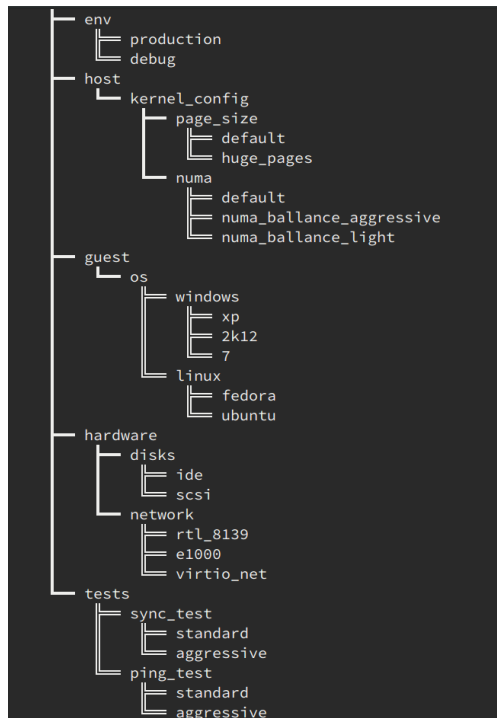
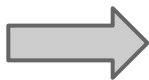
host:
  kernel_config:
    page_size: !mux
    default:
      huge_pages:
        huge_pages: yes
    numa: !mux
    default:
      numa_balance_aggressive:
        numa_balancing: 1
        numa_balancing_migrate_deferred: 32
        numa_balancing_scan_size_mb: 512
      numa_balance_light:
        numa_balancing: 1
        numa_balancing_migrate_deferred: 8
        numa_balancing_scan_size_mb: 32

guest:
  os: !mux
  windows: !mux
  os_type: windows
  xp:
    win: xp
  2k12:
    win: 2k12
  7:
    win: 7

  linux: !mux

```

## 73 line YAML file



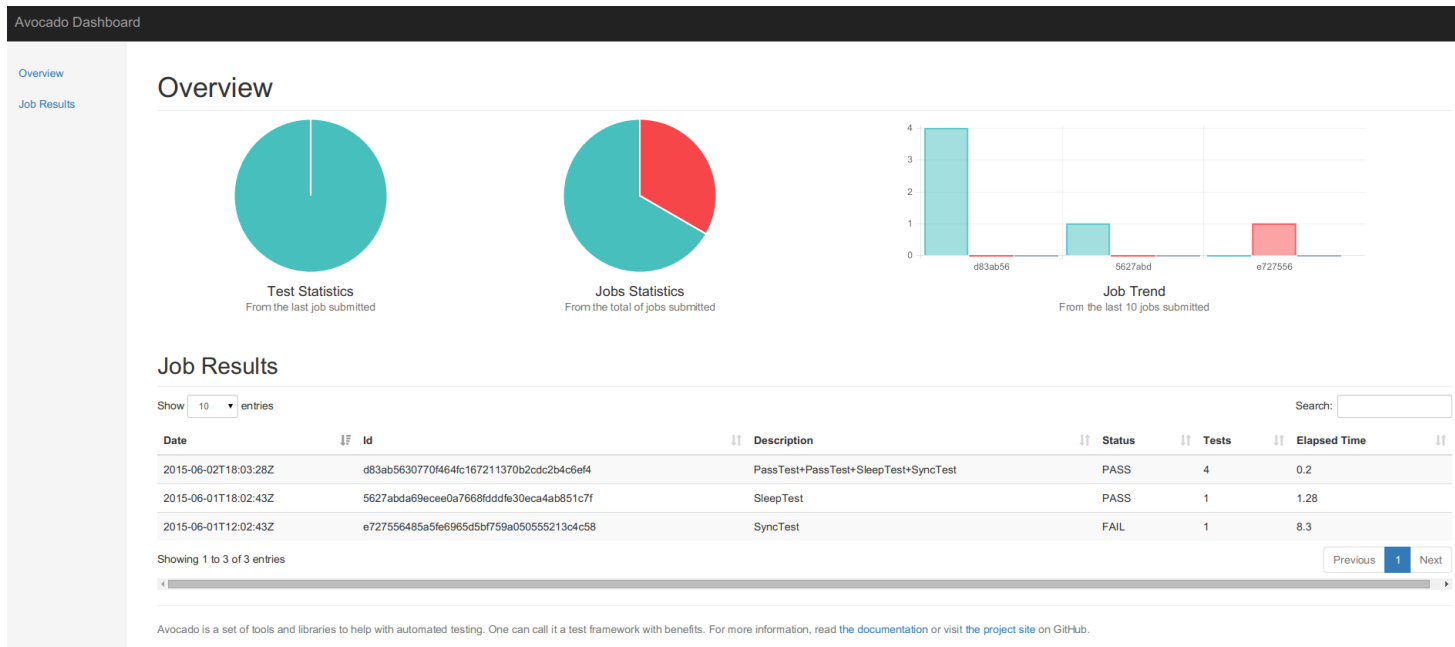
# Tree representation

1440 variants




DEMO  
TIME

# Avocado Dashboard



# Real world: Jenkins Integration

 **Jenkins**  [log in](#)

Jenkins > ARM64-Stable [ENABLE AUTO REFRESH](#)


[Back to Dashboard](#)  
[Status](#)  
[Changes](#)  
[Workspace](#)  
[Build with Parameters](#)  
[Delete Project](#)  
[Configure](#)  
[Rebuild Last](#)  
[Manage Ownership](#)  
[Dependency Graph](#)  
[Schedule Build](#)  
[Job Config History](#)





## Project ARM64-Stable

Run custom avocado/virtest test.

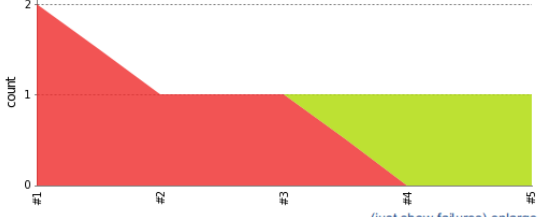
[edit description](#) [Disable Project](#)

Project disk usage information + trend graph

 **Disk Usage:** Workspace 304 KB (On slaves 304 KB, Non slave workspaces -), Builds 3 MB (Locked -), Job directory 3 MB

 [Workspace](#)  
 [Last Successful Artifacts](#)  
 [Recent Changes](#)  
 [Latest Test Result](#) (no failures)

### Test Result Trend













([just show failures](#)) [enlarge](#)


### Permalinks


- [Last build \(#5\), 26 days ago](#)
- [Last stable build \(#5\), 26 days ago](#)
- [Last successful build \(#5\), 26 days ago](#)
- [Last unstable build \(#3\), 28 days ago](#)
- [Last unsuccessful build \(#3\), 28 days ago](#)

### Build History

		<a href="#">trend</a>
 #5	Jul 17, 2015 8:51 AM	
 #4	Jul 17, 2015 8:49 AM	 323 KB
 #3	Jul 15, 2015 4:17 AM	 1 MB
 #2	Jul 14, 2015 1:17 AM	 698 KB
 #1	Jul 8, 2015 1:50 PM	 1 MB

# Real World: Jenkins Integration

**Jenkins**

search  [log in](#)

Jenkins > PPC64LE-Stable > [ENABLE AUTO REFRESH](#)

[Back to Dashboard](#)  
[Status](#)  
[Changes](#)  
[Workspace](#)  
[Build with Parameters](#)  
[Delete Project](#)  
[Configure](#)  
[Rebuild Last](#)  
[Manage Ownership](#)  
[Dependency Graph](#)  
[Schedule Build](#)  
[Job Config History](#)

## Project PPC64LE-Stable

Run custom avocado/virttest test.

[Workspace](#)  
[Last Successful Artifacts](#)  
[Recent Changes](#)  
[Latest Test Result](#) (1 failure / ±0)

[edit description](#)  
[Disable Project](#)

Project disk usage information + trend graph

**Disk Usage:** Workspace 2 MB (On slaves 2 MB, Non slave workspaces -), Builds 38 MB (Locked -), Job directory 38 MB

[Console output](#)

```
Snip! (Full log)
t:Send command: device_add virtserialport,id=vs2,chardev=devvs2,name=vs2
DEBUG:root:(monitor hmp1) Sending command 'device_del vs2'
DEBUG:root:Send command: device_del vs2
DEBUG:root:(monitor hmp1) Sending command 'device_add
virtserialport,id=vs2,chardev=devvs2,name=vs2'
DEBUG:root:Send command: device_add virtserialport,id=vs2,chardev=devvs2,name=vs2
DEBUG:root:(monitor hmp1) Sending command 'device_del vs2'
DEBUG:root:Send command: device_del vs2
INFO:root:[qemu output] qemu-kvm: virtio-serial-bus: Unexpected port id 2 for
device virtio_serial_pci3.0
DEBUG:root:(monitor hmp1) Sending command 'device_add
virtserialport,id=vs2,chardev=devvs2,name=vs2'
DEBUG:root:Send command: device_add virtserialport,id=vs2,chardev=devvs2,name=vs2
DEBUG:root:(monitor hmp1) Sending command 'device_del vs2'
DEBUG:root:Send command: device_del vs2
DEBUG:root:(monitor hmp1) Sending command 'device_add
virtserialport,id=vs2,chardev=devvs2,name=vs2'
DEBUG:root:Send command: device_add virtserialport,id=vs2,chardev=devvs2,name=vs2
INFO:root:[qemu output] qemu-kvm: Guest moved used index from 28475 to 28474
ERROR:root:ThRecvCheck Thread-2: Data loss occurred during socketreconnection.
Maximal loss was 743 per one migration.
DEBUG:root:ThRecvCheck Thread-2: exit(197847)
INFO:root:[qemu output] (Process terminated with status 1)
ERROR:root:Original traceback:
Traceback (most recent call last):
```

### Permalinks

- [Last build \(#16\). 15 days ago](#)
- [Last stable build \(#4\). 29 days ago](#)
- [Last successful build \(#16\). 15 days ago](#)
- [Last failed build \(#2\). 1 mo 2 days ago](#)
- [Last unstable build \(#16\). 15 days ago](#)
- [Last unsuccessful build \(#16\). 15 days ago](#)

Build History		<a href="#">trend</a>
#16	Jul 28, 2015 11:02 AM	2 MB
#15	Jul 28, 2015 10:58 AM	4 MB
#14	Jul 28, 2015 8:33 AM	20 MB
#13	Jul 28, 2015 8:29 AM	6 MB
#12	Jul 28, 2015 8:28 AM	922 KB
#11	Jul 28, 2015 8:23 AM	972 KB

# Avocado: Future

- More external contributions
- Improve virtualization support
- Integrate with more CI tools and provisioning tools
- Avocado server reports and REST tools
- Component isolation (automated bisection)
- ... You decide!



# Resources

- Main website
  - <http://avocado-framework.github.io/>
- Documentation
  - <http://avocado-framework.readthedocs.org/en/latest/>
- COPR repo
  - <https://copr.fedoraproject.org/coprs/lmr/Autotest/>