Machine Learning
finding Linux bugs
Stef Walter, Rado Vbrovsky
Red Hat
“Any sufficiently complex system will have bugs. Bugs are entropy, and entropy is fundamental to this universe.”

— Martin Pitt
Any sufficiently complete testing system will be "plagued" by test flakes.
Cockpit Integration Tests

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Failures</td>
<td>+ 42,764</td>
</tr>
<tr>
<td>Known Issues</td>
<td>+ 54,371</td>
</tr>
<tr>
<td>Test Flakes</td>
<td>+ 25,420</td>
</tr>
</tbody>
</table>

Test Failures: 122,555

Total Runs: 2,466,742

Data from last 90 days:
https://images-cockpit.apps.ci.centos.org/tests-train-1.jsonl.gz
This is your machine learning system?

Yup! You pour the data into this big pile of linear algebra, then collect the answers on the other side.

What if the answers are wrong?

Just stir the pile until they start looking right.
This is a source of training data

```json
{
    "pull": "https://api.github.com/repos/cockpit-project/cockpit/pulls/9230",
    "revision": "2e4fd7d309b7f1c591c099e4041f8db20459c5ae",
    "status": "failure",
    "context": "verify/fedora-27",
    "merged": true,
    "test": "testTeamActive (check_networking_team.TestNetworking)",
    "url": "http://fedorapeople.org/groups/cockpit/logs/pull-9230-20180524-150",
    "tracker": "https://api.github.com/repos/cockpit-project/cockpit/issues/89",
    "log": "...Error: timeout\nwait_js_cond(ph_in_text("#network-interface .p"}
```
## 90+ APIs: File, Command, REST, DBus, Socket

<table>
<thead>
<tr>
<th>API</th>
<th>File</th>
<th>Command</th>
<th>REST</th>
<th>DBus</th>
<th>Socket</th>
</tr>
</thead>
<tbody>
<tr>
<td>abrt</td>
<td>AppStream</td>
<td>apt-get</td>
<td>atomic</td>
<td>Candlepin /candlepin/</td>
<td></td>
</tr>
<tr>
<td>chpasswd</td>
<td>CloudForms</td>
<td>cryptsetup</td>
<td>curl</td>
<td>dbus-daemon</td>
<td></td>
</tr>
<tr>
<td>device-mapper</td>
<td>docker-storage-setup</td>
<td>docker</td>
<td>e2fsprogs</td>
<td>etcd</td>
<td></td>
</tr>
<tr>
<td>/etc/kdump.conf</td>
<td>/etc/passwd</td>
<td>firewalld</td>
<td>FreeBSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSSAPI</td>
<td>hostnamed</td>
<td>ipa-client</td>
<td>ipa-client</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iptables</td>
<td>iscsi-tools</td>
<td>journalctl</td>
<td>kdump</td>
<td></td>
<td></td>
</tr>
<tr>
<td>krb5</td>
<td>Kubernetes /api/</td>
<td>lastlog</td>
<td>libvirt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lvm</td>
<td>mdadm</td>
<td>NetworkManager</td>
<td>NetworkManager-te...</td>
<td>oddjob</td>
<td></td>
</tr>
<tr>
<td>Openshift /oapi/</td>
<td>Openshift OAuth2</td>
<td>openssl</td>
<td>ostree</td>
<td>oVirt /api/</td>
<td></td>
</tr>
<tr>
<td>PackageKit</td>
<td>passwd</td>
<td>PCP</td>
<td>PolicyKit</td>
<td>/proc/meminfo...</td>
<td></td>
</tr>
<tr>
<td>/proc/mounts</td>
<td>/proc/net/dev...</td>
<td>procps-ng</td>
<td>/proc/stat</td>
<td>pwquality</td>
<td></td>
</tr>
<tr>
<td>qemu</td>
<td>realmd</td>
<td>rm ...</td>
<td>rpm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>selinux-policy-target...</td>
<td>selinux-utils</td>
<td>setenforce</td>
<td>Setroubleshootd</td>
<td>shadow-utils</td>
<td></td>
</tr>
<tr>
<td>shutdown</td>
<td>sosreport</td>
<td>ssh-agent</td>
<td>sshd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssh</td>
<td>sssd</td>
<td>storaged</td>
<td>subscription-manager</td>
<td>sudo</td>
<td></td>
</tr>
<tr>
<td>/sys/fs/cgroup</td>
<td>/sys/kernel</td>
<td>/sys/power</td>
<td>systemd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuned</td>
<td>udev</td>
<td>UDisks2</td>
<td>/usr/bin/kubectl</td>
<td></td>
<td>/usr/bin/timedatectl</td>
</tr>
<tr>
<td>/usr/bin/virt-install</td>
<td>/var/log/wtmp</td>
<td>/var/run/utmp</td>
<td>virsh</td>
<td></td>
<td>who/w</td>
</tr>
<tr>
<td>xfsprogs</td>
<td>yum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Testing flakes are those bugs. That entropy.
The tests are essentially fuzzing Linux.
Machine Learning
Cluster the test failures
ML techniques in use

- Preprocessing
- Term Frequency - Inverse Document Frequency
- Normalized Compression Distance
- DBSCAN unsupervised clustering
- Multi-Dimensional Scaling
- K-nearest Neighbors classification
NAME        UUID                                  TYPE            DEVICE
System eth0  5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03  802-3-ethernet  eth0
virbr0       82404faf-cb6b-4f4c-8eb6-65fc1a2ff8da  bridge          virbr0
System eth1  9c92fad9-6ecb-3e6c-eb4d-8a47c6f50c04  802-3-ethernet  --

52:54:01:00:00:03 -> eth2
52:54:01:00:00:03 -> eth2
not ok 139 testTeam (check_networking_team.TestNetworking) duration: 106s
Traceback (most recent call last):
  File "/build/cockpit/bots/../test/verify/check-networking-team", line 81, in 
    b.wait_present("#network-interface-slaves tr[data-interface='%s']" % iface
  File "/build/cockpit/test/common/testlib.py", line 230, in wait_present
    return self.wait_js_func('ph_is_present', selector)
  File "/build/cockpit/test/common/testlib.py", line 224, in wait_js_func
    return self.phantom.wait("%s(%s)" % (func, '',').join(map(jsquote, args))))
  File "/build/cockpit/test/common/testlib.py", line 821, in 
    return lambda *args: self._invoke(name, *args)
  File "/build/cockpit/test/common/testlib.py", line 847, in _invoke
    raise Error(res['error'])
Error: timeout

Wrote TestNetworking-testTeam-rhel-7-127.0.0.2-2601-FAIL.png
Wrote TestNetworking-testTeam-rhel-7-127.0.0.2-2601-FAIL.html
Journal extracted to TestNetworking-testTeam-rhel-7-127.0.0.2-2601-FAIL.log
Journal extracted to TestNetworking-testTeam-fedora-26-127.0.0.2-2602-FAIL.log
Extracting the logs

```
# testTeam (check_networking_team.TestNetworking)
#
# NAME         UUID                                  TYPE            DEVICE
System eth000  xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  000-000-ethernet  eth000
virbr000       xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  bridge          virbr000
System eth000  xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  000-000-ethernet  --

000:000:000:000:000:000 -> eth000
000:000:000:000:000:000 -> eth000
not ok
Traceback (most recent call last):
File "check-networking-team", line 000, in testTeam
b.wait_present("#network-interface-slaves tr[data-interface='%s']" % iface000)
File "testlib.py", line 000, in wait_present
return self.wait_js_func('ph_is_present', selector)
File "testlib.py", line 000, in wait_js_func
return self.phantom.wait("%s(%s)" % (func, ','.join(map(jsquote, args))))
File "testlib.py", line 000, in
return lambda *args: self._invoke(name, *args)
File "testlib.py", line 000, in _invoke
raise Error(res['error'])
Error: timeout

Wrote png
Wrote html
Journal extracted to log
Journal extracted to log```
Term Frequency - Inverse Doc Frequency

```python
# testTeam (check_networking_team.TestNetworking)
virbr000  xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  bridge  virbr000
File "check-networking-team", line 000, in testTeam
b.wait_present("#network-interface-slaves tr[data-interface='%s']" % iface000)
```
Normalized Compression Distance

\[ NCD_Z(x, y) = \frac{Z(xy) - \min\{Z(x), Z(y)\}}{\max\{Z(x), Z(y)\}}. \]

\[ Z = \text{lambda } v: \text{len(lz4.frame.compress(v))} \]
DBSCAN

Density-based spatial clustering of applications with noise
Multi-Dimensional Scaling
k-nearest neighbor
Make it do something
Auto-retrying flakey tests


Result directory
Raw log

Done: 1 tests failed.

247 tests, 224 passed, 1 failed, 22 skipped, -3 to go (3 retries).
Failed and skipped tests:

13. testPtp (check_metrics.TestMetrics) # duration: 47s # RETRY due to flakiness [screenshot] [journal]
23. testBondingMain (check_networking_bond.TestNetworking) # duration: 88s # RETRY due to flakiness [screenshot] [journal] [journal]
23. testBondingMain (check_networking_bond.TestNetworking) # duration: 88s # RETRY due to flakiness [screenshot] [journal] [journal]
224. testProjectGroups (check_openshift.TestRegistryPrerelease) # duration: 299s [screenshot] [journal] [journal]

1: testBasic (check_kdump.TestKdump) [screenshot] [journal] -- skipped: Known Issue #9993
3: testDashboard (check_kubernetes.TestKubernetes) [screenshot] [journal] -- skipped: Known issue #10011
7: testTopoology (check_kubernetes.TestKubernetes) [screenshot] [journal] -- skipped: Known Issue #10011
9: testDeployDialog (check_kubernetes.TestNucleus) -- skipped: Nucleus deploy temporarily removed
39: testTeam (check_networking_team.TestNetworking) [screenshot] [journal] [journal] [journal] [journal] [journal] -- skipped: Known Issue #6108
64: testVdoMissingPackages (check_storage_vdo.TestStoragePackages) -- skipped: No vdo available
80: testRegister (check_subscriptions.TestSubscriptions) -- skipped: Fedora-29: No subscriptions
86: testTimeservers (check_system_info.TestSystemInfo) [screenshot] [journal] -- skipped: Known issue #9994
94: testExternalConsole (check_machines_vrsh.TestMachinesVrsh) -- skipped: Broken with current chromium, see PR #9229
125: testExternalConsole (check_machines_vrsh.TestMachinesVrsh) -- skipped: Broken with current chromium, see PR #9229
140: testInfoSecurity (check_packagekit.TestUpdates) [screenshot] [journal] -- skipped: Known Issue #9996
141: testInfoTruncation (check_packagekit.TestUpdates) [screenshot] [journal] -- skipped: Known Issue #9996
143: testSecurityOnly (check_packagekit.TestUpdates) [screenshot] [journal] -- skipped: Known Issue #9996
147: testAvailableUpdates (check_packagekit.TestUpdatesSubscriptions) -- skipped: Fedora-29: No subscriptions
Annotate test logs

```
# testSuper (check_reauthorize.TestReauthorize) # duration: 25s

DevTools listening on ws://127.0.0.1:8082/devtools/browser/20132904-84e8-492a-bda1-07659595e814
[0123/152927.826842:ERROR:zygote_host_impl_linux.cc(266)] Failed to adjust DOM score of renderer with pid 129804: Permission denied (13)
[0123/152927.864472:ERROR:zygote_host_impl_linux.cc(266)] Failed to adjust DOM score of renderer with pid 129801: Permission denied (13)
> log: done
> log: fail
Traceback (most recent call last):
  File "/build/cockpit/outs/0/test/verify/check-reauthorize", line 84, in testSuper
    self.assertEqual(s.text("-super-channel span"), 'result: access-denied')
AssertionError: u'result: disconnected' != 'result: access-denied'

not ok 85 testSuper (check_reauthorize.TestReauthorize) # duration: 25s
```

Wrote screenshot to TestReauthorize-testSuper-fedora-1386-127.9.0.22601-FAIL.png
Wrote HTML dump to TestReauthorize-testSuper-fedora-1386-127.9.0.2-2601-FAIL.html
Wrote JS log to TestReauthorize-testSuper-fedora-1386-127.9.0.2-2601-FAIL.js.log
Journal extracted to TestReauthorize-testSuper-fedora-1386-127.9.0.2-2601-FAIL.log

# Flake probability: 72.5% (neural network)
# Flake likely 66.2% (clustering)
Curated set of non-flakey tests

0 verify/fedora-27 testBasic (check_ovirt.TestOVirtMachines)
0 verify/fedora-27 testBond (check_networking_bond.TestNetworking)
0 verify/fedora-27 testDeployDialog (check_kubernetes.TestNulecule)
0 verify/fedora-27 testDeployDialog (check_openshift.TestOpenshift)
0 verify/fedora-27 testDeployDialog (check_openshift.TestOpenshiftPrep)
0 verify/fedora-27 testDockerCommandInfo (check_openshift.TestRegistry)
0 verify/fedora-27 testExternalConsole (check_machines.TestMachines)
0 verify/fedora-27 testIpa (check_realms.TestRealms)
0 verify/fedora-27 testKubevirtMachinesCreate (check_openshift.TestOpenshift)
0 verify/fedora-27 testLuks (check_storage_luks.TestStorage)
0 verify/fedora-27 testPendingClaim (check_kubernetes.TestKubernetes)
0 verify/fedora-27 testRaid (check_storage_mdraid.TestStorage)
50 verify/fedora-27 testBasic (check_apps.TestApps)
50 verify/fedora-27 testBasic (check_storage_mdraid.TestStorage)
50 verify/fedora-27 testHostToMaintenance (check_ovirt.TestOVirtMachines)
50 verify/fedora-27 testKubevirtMachinesList (check_openshift.TestOpenshift)
50 verify/fedora-27 testProjectGroups (check_openshift.TestRegistryPrep)
60 verify/fedora-27 testBasic (check_kdump.TestKdump)
100 verify/fedora-27 testCheckpoint (check_networking_checkpoints.TestNetworking)
100 verify/fedora-27 testDosParts (check_storage_msdos.TestStorage)
100 verify/fedora-27 testISCSI (check_storage_iscsi.TestStorage)
100 verify/fedora-27 testKubevirtMachinesCreate (check_openshift.TestOpenshift)
100 verify/fedora-27 testKubevirtMachinesList (check_openshift.TestOpenshift)
100 verify/fedora-27 testKubevirtVmInstance (check_openshift.TestOpenshift)
100 verify/fedora-27 testPendingClaim (check_openshift.TestOpenshift)
100 verify/fedora-27 testPendingClaim (check_openshift.TestOpenshiftPrep)
100 verify/fedora-27 testProjectPolicy (check_openshift.TestRegistry)
100 verify/fedora-27 testUndeployKubevirt (check_openshift.TestOpenshift)
File issues for "big" bugs

testDockerCommandInfo (check_openshift.TestRegistry) is flaky #10845

cockpituous commented on Dec 17, 2018

86% on verify/fedora-29

- pull-10702
- pull-10638

Source material

cockpituous added the flake label on Dec 17, 2018
Visualization
Real bugs

- **Example:** Shadow utils usage races
  
  cluster-7-112.log  
  https://github.com/cockpit-project/cockpit/pull/9979

- **Example:** Networkd + SELinux breaks the boot
  
  cluster-124-4.log

- **Example: Race invalid update of UI state**
  
  cluster-95-28.log  
  https://github.com/cockpit-project/cockpit/pull/10127

- **Example:** PackageKit crashing intermittently
I want!
https://github.com/cockpit-project/cockpituious/tree/master/learn
Use it yourself

$ git clone git@github.com:cockpit-project/cockpituous
$ cd cockpituous

$ ./train-tests -v test-example.jsonl.gz
Loading existing tests data
14208: Items to train
...
100933632: Computed distances in 1181 seconds on 32 cores
332: Clusters (13962 items, 246 noise)
$ ./predict-tests -v test-predict.jsonl
{
  "test": [ [ "testTeamActive", 2 ], [ "testTeamBond", 5 ] ],
  "status": [ [ "failure", 7 ] ],
  "context": [ [ "verify/fedora-27", 3 ], [ "verify/fedora-27", 4 ] ],
  "merged": [ [ true, 5 ], [ false, 1 ], [ null, 1 ] ],
}
Use it with Kubernetes

$ kubectl create -f learn/cockpit-learn.yaml

$ curl --progress-bar --fail --upload-file learn/test-example.jsonl.gz
  http://cockpit-learn.project.srv.cluster.local/train/test-example.jsonl.gz

$ curl -d '@learn/test-predict.jsonl'
  http://cockpit-learn.project.srv.cluster.local/predict
What's next

- Retrieving data automatically
- Tracking known issues automatically
<table>
<thead>
<tr>
<th>ID</th>
<th>Data Dir</th>
<th>Case no.</th>
<th>Tracker no.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Provided calltrace</th>
<th>Suggested calltrace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 dump_stack</td>
<td>1 dump_stack</td>
</tr>
<tr>
<td>2 report_bad_irq</td>
<td>2 report_bad_irq</td>
</tr>
<tr>
<td>3 note interrupt</td>
<td>3 note interrupt</td>
</tr>
<tr>
<td>4 handle irq event percpu</td>
<td>4 handle irq event percpu</td>
</tr>
<tr>
<td>5 handle irq event</td>
<td>5 handle irq event</td>
</tr>
<tr>
<td>6 handle fasteoi_irq</td>
<td>6 handle fasteoi_irq</td>
</tr>
<tr>
<td>7 handle irq</td>
<td>7 handle irq</td>
</tr>
<tr>
<td>8 rtqueue_account_irq</td>
<td>8 sched_clock_idle_wakeup_event</td>
</tr>
<tr>
<td>9 do IRQ</td>
<td>9 do IRQ</td>
</tr>
<tr>
<td>10 common interrupt</td>
<td>10 common interrupt</td>
</tr>
<tr>
<td>11 cpuidle enter state</td>
<td>11 cpuidle text start</td>
</tr>
<tr>
<td>12 cpuidle enter</td>
<td>12 cpuidle text start</td>
</tr>
<tr>
<td>13 call cpuidle</td>
<td>13 native safe halt</td>
</tr>
<tr>
<td>14 do idle</td>
<td>14 default_idle</td>
</tr>
<tr>
<td>15 cpu_startup_entry</td>
<td>15 arch_cpu_idle</td>
</tr>
<tr>
<td>16 rest_init</td>
<td>16 cpu_startup_entry</td>
</tr>
<tr>
<td>17 start kernel</td>
<td>17 rest_init</td>
</tr>
<tr>
<td>18 start kernel</td>
<td>18 start kernel</td>
</tr>
<tr>
<td>19 generic conv string</td>
<td>19 generic conv string</td>
</tr>
</tbody>
</table>
Ludicrous Claim: Any testing system that does not have enough flakes, should introduce more tests or more mutations.
Questions?

cockpit-project.org

#cockpit on FreeNode

Credits:
Clusters: Chire on Wikipedia
CC BY-NC 2.5: Comic by XKCD.com