Better development with robots

Aaron Conole <aconole@redhat.com>
September 19, 2019
My presentation is a lot like yours, only more interesting ‘cause it involves robots

The talk

- Robot here is code for CI enabling technologies
- Q: What robots are scurrying about when you post a patch?
- Q: How can I avoid being taunted by a machine?
- Q: I want to help out with robots, how do I do that?
- Community feedback
Hey developers, wanna test all patches?

Patches are public
- Everyone submits them
- People don’t want to admit that they fail to test them
- We can all do better

Tests can be run on the patches
- We know what got submitted
- Just build it!
- Knowing where to apply the patch can be tricky
- Please - run the unit tests!

Reports are public
See http://mails.dpdk.org/archives/test-report/
Oh, I get it— make the robot do all the work.

All these patches, like tears in a rain
- Developers come up with a bunch of patches
- They send them via email to the mailing list
- Patchwork gets those emails and builds up series information
- The robots monitor patchwork and start processing
Malfunction! Need input!

How a patch gets to a robot
Here I am with a brain the size of a planet

http://patches.dpdk.org
How can I be so bad at everything I try and still be so great

Sometimes not everything passes

[v4,01/54] ethdev: remove redundant device in...
Test my shiny metal patch

<table>
<thead>
<tr>
<th>How can a developer get some of this right</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ It’s great that these safeguards exist</td>
</tr>
<tr>
<td>▪ Better to get it right on the first try</td>
</tr>
<tr>
<td>▪ You can still be ’lazy’ though</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GitHub + Travis</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Sign up to github</td>
</tr>
<tr>
<td>▪ Link it to travis-ci</td>
</tr>
<tr>
<td>▪ Push your changes</td>
</tr>
<tr>
<td>▪ Profit!</td>
</tr>
</tbody>
</table>
Thanks for your cooperation.
sys_stv - does not compute

What It does

- Build code
- Intel specific
- Various compiler and OS configurations

Table: SYS-STV Build Matrix sample

<table>
<thead>
<tr>
<th>os</th>
<th>Makefile gcc</th>
<th>Makefile clang</th>
<th>...</th>
<th>Meson gcc/clang</th>
<th>Docs pdf/html</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHEL76-64</td>
<td>pass</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CENTOS76-64</td>
<td>pass</td>
<td>pass</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UB1804-64</td>
<td>pass</td>
<td>pass</td>
<td>...</td>
<td>pass</td>
<td>pass</td>
</tr>
<tr>
<td>UB1604-32</td>
<td>pass</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Red Hat Robot - scaring the life out of you

What it does
- Push a series to a special branch
- Monitors travis-ci run
- Discards all the data and runs into walls

No reports?
- Need to fix some of the sporadic failures
UNH IOL

Overview
- Neutral 3rd party
- Hosts setups from Intel, MLX, Broadcom, NXP
- Dedicated staff (thanks, Jeremy!)

Infra
- Traffic generators, and devices under test
- Dashboard! https://lab.dpdk.org/results/dashboard/
- SPDK smoke tests
- OvS perf tests https://github.com/chaudron/ovs_perf/

Perf Tests
- Tuning difficulties (baremetal v. virt, hw, sw, fw combinations)
- DUT tuning vs Traffic Gen tuning
IOL Screenshot

http://lab.dpdk.org

Tarball 6130

Tarball Info
dpdk (f991c135b35af6d34fe8f06b20c9e92e1d2fa4)

Date created
Sept. 12, 2019, 9:46 p.m.

Test Results
Possible Regression

Ubuntu 18.04 VM

Configuration Information

Kernel      linux 4.15.0-generic
Compiler    gcc 7.4

dpdk_compile
Directive 1 - Serve the public trust

Get involved
- Most of the steps are stored in .ci/ subdir
- Makes builds, runs unit tests
- Simple scripts

What still remains
- Reporting from the RH robot
- aarch64 unit tests (utilize qemu-user)
- Maybe ppc build
- Code coverage reporting (to identify gaps)
- Other test suites
Beyond Travis - GitLab, Actions, Cirrus, AppVeyor

**Other public automation technologies exist**
- Cirrus supports Windows and FreeBSD testing
- GitLab can run in a 'hybrid' mode
- Actions is new from Github (maybe for better pull request?)

**Build your own**
- Everything doable with patchwork APIs
- [https://github.com/orgcandman/pw-ci](https://github.com/orgcandman/pw-ci) - For monitoring patchwork
Questions