LinuxCon Europe
UEFI Mini-Summit
7 October 2015

Session 3 – LUV Shack: An Automated Linux Kernel and UEFI Firmware Testing Infrastructure
Matt Fleming, Intel
Linux* UEFI Validation Project

• Started in January 2014
• Custom Linux Distribution (Yocto Project)
• Provides boot time and runtime testing
• Crash fast, crash hard, then recover
LUV Project
The 5000 mile problem

- USB live images do not scale
- Geography is a barrier
- Debugging vicariously is hard
LUV Shack

- Buildbot used as controlling mechanism
  - Continuous integration framework
  - Written in Python
- Mixture of physical and virtual machines
- Everything is version controlled
LUV Shack

REPOSITORY
- Subversion
- Mercurial
- Bazaar
- Darcs
- GIT
- CVS

BUILD MASTER

NOTIFIERS
- Email
- Web Status
- IRC
- Status Client

BUILD SLAVE

BUILD SLAVE
LUV Shack - overview

GitHub

Poll

Changes

BUILD MASTER

Status

Aarch64 BUILD SLAVE

IA32 BUILD SLAVE

X64 BUILD SLAVE
LUV Shack - changes

GitHub

"master"

Everything is a git branch
factory = BuildFactory()
factory.addStep(Git(repourl='git://github.com/01org/luv-yocto.git',
    branch='next', mode='full'))
factory.addStep(ShellCommand(description="build image",
    command=['/data/buildbot/luv-build', "next"]))
factory.addStep(Trigger(schedulerNames=['next-tests-x64'],
    waitForFinish=True))
LUV Shack - results

[+] test_create_read... passed
[+] test_delete... passed
[+] test_zero_size_delete... passed
[+] test_open_unlink... passed
[+] test_valid_filenames... passed
[+] test_invalid_filenames... passed
[-] fwts
  [+] bios_info... passed
  [+] version... passed
  [+] acpiinfo... passed
  [+] mtrr... passed
  [+] klog... passed
  [+] oops... passed
  [+] acpitable... passed
  [+] apicinstance... passed
  [+] asf... passed
  [+] bert... passed
  [+] bgrt... passed
  [+] boot... passed
  [+] autobrightness... passed
  [+] checksum... passed
LUV Shack Wins

- Can run tests in parallel
- Debug issues remotely
- Plug and play testing
- Test out kernel and firmware patches easily
- Regression testing
- Orders of magnitude improvement in development time
LUV Shack Evolution

1. Running on developer’s machines
LUV Shack Evolution

1. Running on developer’s machines
2. Run LUV on remote machines
LUV Shack Evolution

1. Running on developer’s machines
2. Run LUV on remote machines
3. Build custom LUV and deploy in parallel
Case Study

- EFI_PROPERTIES_TABLE new in UEFI v2.5
- Required Linux kernel enabling
- We didn’t have access to hardware
- Solution? A new luv-yocto.git branch!
Final Thoughts

• Please run LUV and report results
  – USB image
  – PXE
  – Qemu
• Tell us about your tests
• LUV integrates with other validation tools
Resources

• luv@lists.01.org
• https://01.org/linux-uefi-validation
• https://github.com/01org/luv-yocto
• http://docs.buildbot.net/current/tutorial/
• https://www.yoctoproject.org
Questions?
Interested in Joining?
www.uefi.org/membership

UEFI FW/OS Forum:
uefi.org/FWOSForum
A free public forum focused on firmware and O/S integration

USRT Security Issue Reporting:
uefi.org/security
A safe reporting site to inform the UEFI of any security issue or vulnerability based on firmware