OpenXT:
Test Automation in OpenXT
BVT & ATF

Garrett Morgan

Chris Rogers

AIS, Inc.

June 7, 2016

Briefing Overview

Test Automation

devastating capability, revolutionary advantage

- Background
- Design & Infrastructure
- Capabilities
- Current Status
- Improvement Opportunities
- Looking Forward

Background - BVT

Test Automation

devastating capability, revolutionary advantage

- **Build Validation Testing**
  - https://github.com/OpenXT/bvt
- **Functions both as test & task automation framework**
- **Specifically written at Citrix for XenClientXT based on parts of XenServer RT [Regression Testing]**
  - Commit on 6/30/15 to provide base functionality to OXT by Chris
  - Currently being maintained at AIS by Chris, Garrett & Interns
- **Uses ssh to perform remote command execution on test platforms.**

Design - BVT

Test Automation

- **BVT defines a testing library and a series of testcases**
  - Intuitive directory structure
  - Makes use of dynamic module loading
- **Utilizes two main control scripts**
  - bvt.py: CLI for testcase execution
  - autolaunch.py:
    - Automation wrapper for running test suites (one or more test cases)
    - Also responsible for processing scheduled tests through the UI.
- **Utilizes two helper daemons**
  - bvt_daemon
    - Monitors the job queue for scheduled tests
  - build_watcher
    - Monitors specified buildbot instances for new builds

Design – BVT (cont.)

Testing Automation

- **BVT also makes use of a variety of support tools**
  - mongodb
  - pxe
  - tftp
  - dhcp
  - wsman
  - amttool
  - private_settings

Infrastructure - BVT

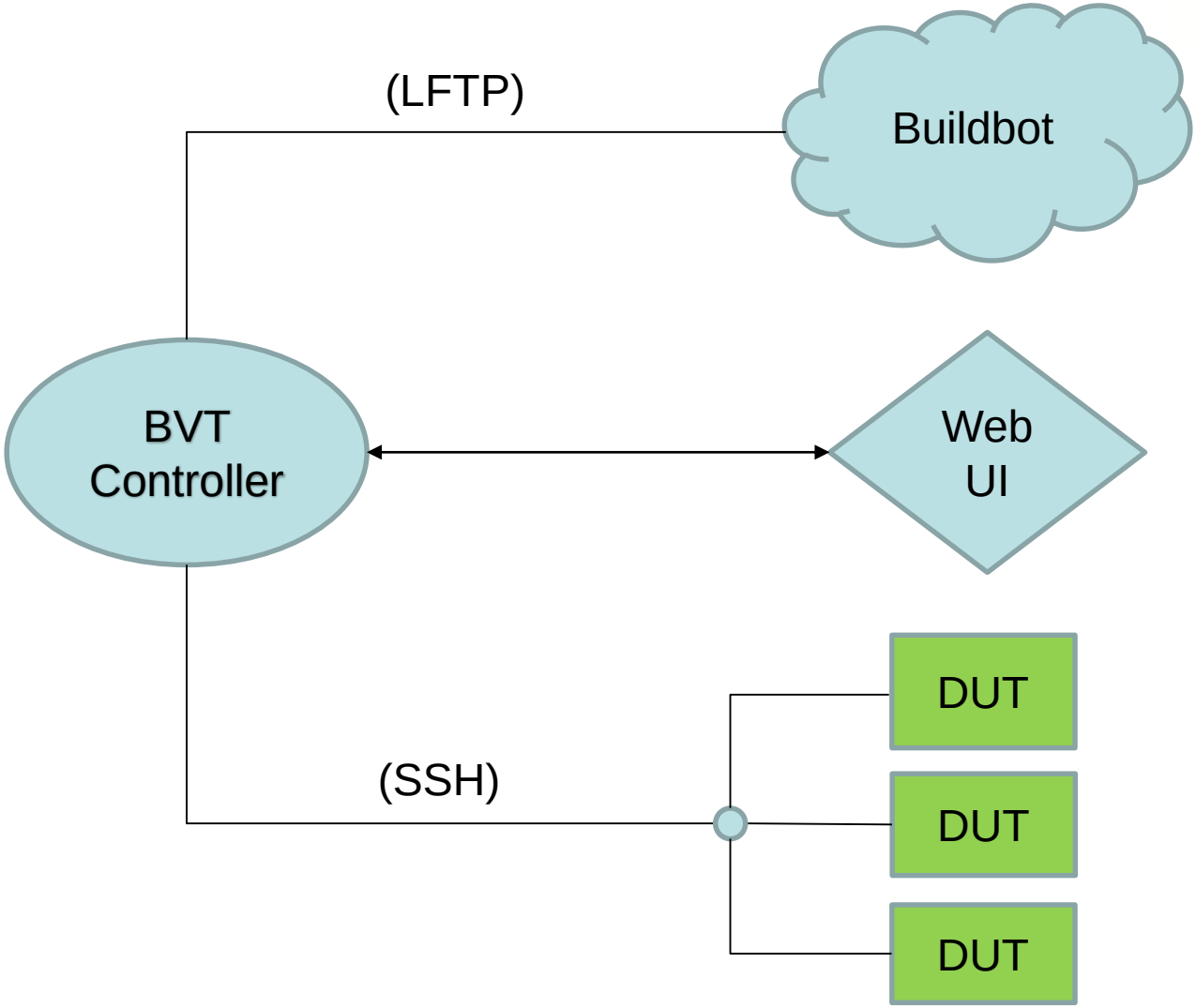
Test Automation

- **Controller**
  - Debian workstation
  - Private LAN with dhcp/PXE
    - Ability to query for VM IP addresses and serve PXE images for hosts/VMs
  - AMT control
    - Uses amttool or wsman depending on AMT version
- **DUTs (Devices under Test)**
  - OXT-compatible workstations with AMT
  - Dual-NIC
    - Onboard AMT-enabled NIC
    - Add-on NIC for NDVM/dom0 connectivity
    - Both interfaces connected to private BVT controller private LAN
  - Controller communicates with guests via ssh or exec_daemon

Infrastructure – BVT

Test Automation

devastating capability, revolutionary advantage



Capabilities – BVT

Test Automation

- **Current capabilities**
  - Test/task automation and scheduling
  - BuildBot integration
  - Reporting
    - Zephyr (JIRA) integration
    - webUI
    - Stdout
- **Upstream Current testcases**
  - boot_time – clocks startup time
  - check_mac_addrs – Ensures mac addresses are correct
  - install_guest – Installs a guest from vhd or iso
  - install_tools – Automatically install PV tools
  - pxe_install_xc – Install a specified OXT build
  - reboot_test – Reboot the host for a specified duration

Current Status – BVT

Test Automation

devastating capability, revolutionary advantage

- **BVT testbeds**
  - AIS – Rome, NY
    - VMWare virtualized controller & ~6 DUTs
  - OpenXT – Burlington, MA
    - Primary test focus of OpenXT
    - Controller/DUT stats
    - Implementation led to creation of OXT-550
- **Still under development**

Current Status – BVT

Test Automation

devastating capability, revolutionary advantage

- **Issues found during BVT test development**
  - 344 – handle 4 disk slot limitation in UI
  - 348 – UI does not allow user to set vCPUs above hardware max
  - 351 – UIVM does not function if its domid is greater than 254
  - 354 – issuing shutdown/reboot to S3 guest results in inconsistent behavior
  - 356 – sound card controls listed for non-stubdom VMs that are shutdown
  - 382 – MAC address translation for NDVM/Dom0 is not consistent
  - 406 – NDVM with add-on NIC only will not get IP address after NDVM reboot

Current Status – BVT

Test Automation

- **Pros**
  - Code base is easy to read/traverse
  - Finds bugs and provides automation to look for bugs in ways that normal procedural testing would not
  - Provides sanity tests for new builds coupled with build watching
- **Areas for Improvements**
  - New users to help work through documentation and installation issues
  - Hardware price of admission
  - Simpler install process
  - Identifying and squashing bugs

Looking Forward – BVT

Test Automation

devastating capability, revolutionary advantage

- **Provide OXT-based VM for a BVT Controller**
  - Possible opportunity for a service VM
- **Re-write of BVT as OO**
- **Automated TXT/TPM reset for MLE installs**
- **Reduce complexity of bvt/autolaunch/daemon architecture**
- **Remove non-working tests & libraries**
- **Continually reduce ease/cost of entry**

Background - ATF

Test Automation

- Automated Test Framework
- Initial research into an UI-Automation framework for OXT-based platforms
- Development initially done in 2013/2014 and restarted in 2016 to provide proof of concept
- Uses SikuliX and AMT KVM

Design – ATF

Test Automation

- **Controller**
  - Windows 7
    - SikuliX (python)
    - RealVNC Viewer+ (AMT-enabled)
- **DUT**
  - AMT-KVM enabled OXT-compatible workstation
  - Additional NIC for NDVM
- **OO Design:**
  - VM class
    - VM.openVMDetails(), VM.changevCPUs(), VM.changeMemory()
  - ThickVM class
    - Inherited from VM
  - Host class
    - Host.runTerminalCommand(), Host.restart(), Host.settings()

Capabilities – ATF

Test Automation

- **Provides a DUT closer to an end-user installed workstation**
- **Proof of concept testcases using framework**
  - MLE install via PXE or CD for Dell Platforms
    - nonMLE installs for other platforms
  - VM creation
  - Interactive Windows Installation
  - Windows Tools Installation
  - Most UIVM functionality

Current Status - ATF

Test Automation

- **Integration level with BVT**
- **Continue to focus on functional and regression testing on OXT platforms.**
  - Only use a UI framework where testing of the product is needed, eliminating for setup of tests where possible.

Looking Forward - ATF

Test Automation

- **Grow test library based on architecture decision**
  - Include other OS installers & increase flexibility of existing ones
    - Linux -> Debian, Ubuntu, RHEL
    - Windows10
  - Include tests for any UI specific issues

End

Test Automation

devastating capability, revolutionary advantage

- **Thanks for listening!**