OSFCI : Open Firmware CI with real hardware execution.

Jean-Marie Verdun, Distinguished Technologist
Arun Darlie Koshy, Senior Engineer

HPE, Advanced Technology Team

https://github.com/opencomputeproject/OSF-OSFCI
FROM
The pandemic hit hard
- Firmware development had to become fully remote

Dual security model
- Open-source with customer control
- Proprietary with supplier liability

Enable adoption at scale
GOALS

• Automatize remote build, testing and release management of BMC and ROM firmware

• Validate on target that Open-Source Firmware works as expected (OpenBMC and LinuxBoot)

• Have specially provisioned instances for customers, public and OCP

• Make it fully open source and encourage community adoption
Open Source Firmware at scale

Welcome to the development platform dedicated to implementation and testing of Open Source FW on HPE Proliant/Apollo platforms
DEMO – BUILD OPENBMC

Just drag and drop your BMC firmware here.

Just drag and drop your standard bios file here.

Or

Specify a github repo and branch.

Or

unkoshy/openbmc.git pedl

load standard OpenBMC

build

Load my Firmware

DL

end session

arunkoshy
DEMO – LOAD

Just drag and drop your BMC firmware here.

Specify a GitHub repo and a branch.

Load standard BMC

load standard OpenBMC

Or

https://github.com/arunkosh

build

Load my Firmware

DL

Sent 29968128 bytes of 33554432
Sent 32457200 bytes of 33554432
Sent 33554432 bytes of 33554432
Transfer complete

Read 269732 bytes of 33554432
Read 492542 bytes of 33554432
Read 599515 bytes of 33554432
Read 238693 bytes of 33554432
Read 488560 bytes of 33554432
Read 1668864 bytes of 33554432
Read 19977210 bytes of 33554432
Read 10873368 bytes of 33554432
Read 28975260 bytes of 33554432
Read 21862472 bytes of 33554432
Read 13249042 bytes of 33554432
Read 3948820 bytes of 33554432
Read 3189576 bytes of 33554432
Read 2005452 bytes of 33554432
Verify: PASS

Starting DLWPPro. Press CTRL-C to exit.
WHY IS THIS CRITICAL?

• Easier developer workflow around build, debug and observability on real hardware

• Manual firmware development and validation has been disrupted by hybrid / remote-work model

• Enhanced hardware safety
ARCHITECTURE

- Modular architecture and configurable
- Written in Go

- API endpoints supporting:
  - Automation including external frameworks like Contest
  - Non-interactive mode
  - Build and load firmware, log retrieval, power operations
  - Example(s):
    - https://github.com/opencomputeproject/OSF-OSFCI/tree/master/api

- Enhanced analytics and logging
- Fully open-source under MIT license
OUTCOMES

• In use externally and internally
  • For both proprietary and open-source development

• Seed community efforts around a common set of tests

• Combine innovations from community members

• Used by OCP (Open Compute Project)