OPEN

Compute Project

ACPI Platform Description (APD)

Contents

Executive Summary

In Open Networking, the disaggregation of switching hardware from the network operating system (NOS) has created tremendous value for consumers by giving them the freedom of choice. However, the integration of the NOS with platform-specific components on bare metal switches is an unpredictable endeavor requiring device vendors to work closely with NOS vendors months before a hardware/software combination is considered credible by the market. Essentially, the hardware vendor is at the mercy of the NOS vendor to support the platform. The PC industry faced a similar battle in the early 90s. As such, hardware and software vendors banned together to create the Advanced Configuration and Power Interface (ACPI) which allows hardware vendors to describe their platform in a machine readable format that an operating system (OS) can use to discover platform-specific hardware details. It was this subsystem that allowed hardware vendors to ship hardware on their schedule, software vendors to support new hardware they haven't seen, and gave consumers an expectation that PC hardware platforms and operating systems should just work.

ACPI Platform Description (APD) aims to leverage the same lessons learned and bring ACPI to the network industry. Hardware vendors will bring products to market faster, NOS vendors will support more hardware platforms, and consumers will benefit from more choice in terms of both hardware and software.

Revision History

Author	Date	Version	Description
Dustin Byford	3/31/2015	0.1	Initial Release

Overview

In Open Networking, the disaggregation of switching hardware from the network operating system (NOS) has created tremendous value for consumers by giving them the freedom of choice. However, the integration of the NOS with platform-specific components on bare metal switches is an unpredictable endeavor requiring device vendors to work closely with NOS vendors months before a hardware/software combination is considered credible by the market. Essentially, the hardware vendor is at the mercy of the NOS vendor to support the platform. The PC industry faced a similar battle in the early 90s. As such, hardware and software vendors banned together to create the Advanced Configuration and Power Interface (ACPI) which allows hardware vendors to describe their platform in a machine readable format that an operating system (OS) can use to discover platform-specific hardware details. It was this subsystem that allowed hardware vendors to ship hardware on their schedule, software vendors to support new hardware they haven’t seen, and gave consumers an expectation that PC hardware platforms and operating systems should just work.

ACPI Platform Description (APD) aims to leverage the same lessons learned and bring ACPI to the network industry. Hardware vendors will bring products to market faster, NOS vendors will support more hardware platforms, and consumers will benefit from more choice in terms of both hardware and software.

APD achieves these goals by defining a standard way to describe network switch components using ACPI, implementing that standard in Linux kernel drivers, and providing a set of tools and documentation to lower the barrier for hardware vendors to describe their platforms using APD. The work is all in the process of being contributed to relevant open source projects and standards bodies such as the Linux Kernel, the UEFI forum, and OCP. With an open standard and implementation, all Open Networking platforms and OSes will benefit.

License

The source and sample files for apd-tools is released under the MIT open source license defined at http://opensource.org/licenses/MIT . All other source is released under respective project open source licenses:

- The Linux Kernel’s GPL license https://www.kernel.org/pub/linux/kernel/COPYING

Background

See Figure 1 as an illustration of a NOS vendor supporting a new hardware platform.

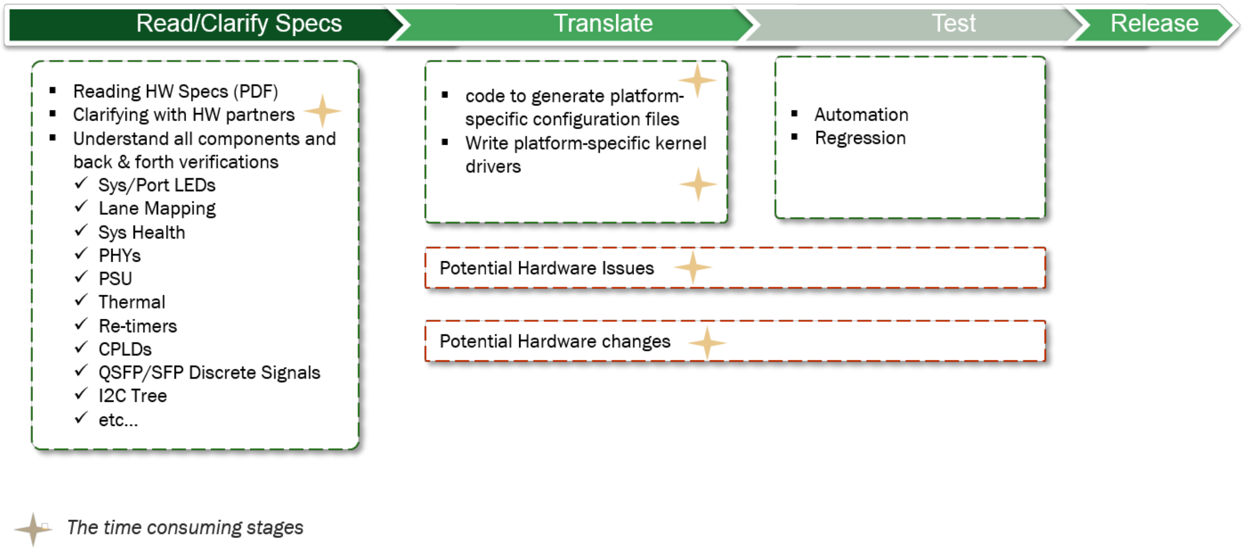
Read/Clarify Specs
Translate
Test
Release
Reading HW Specs (PDF)
Clarifying with HW partners
Understand all components and back & forth verifications
✓ Sys/Port LEDs
✓ Lane Mapping
✓ Sys Health
✓ PHYs
✓ PSU
✓ Thermal
✓ Re-timers
✓ CPLDs
✓ QSFP/SFP Discrete Signals
✓ I2C Tree
✓ etc...
code to generate platform-specific configuration files
Write platform-specific kernel drivers
Automation
Regression
Potential Hardware Issues
Potential Hardware changes
The time consuming stages

Figure 1 - Process to support a new hardware platform

In the first step of supporting a new platform, the vendor must read a set of documents that describe the components of the device. Typically the document is not in sync with the current iteration of the hardware which adds to the delay of understanding the hardware. Once a solid understanding of the platform description has been achieved, coding will start to develop platform specific kernel drivers and generate configuration files for the platform. It is during initial code development and testing will it is normal to see issues crop up, hardware or software. If hardware changes are needed, they are also occurring during this stage as well, increasing the time to market for both vendors. As denoted, it is easy to see how this process is highly variable and non-standard in terms of time and predictability.

As stated previously, this is the same problem the PC industry was faced with in the 90s with their answer being ACPI. Since that time, the ACPI standard has received numerous updates and increased adoption throughout the industry. Given ACPI’s market penetration, it is a natural choice to use and extend it to support network devices. The end result for the consumer is best captured in Figure 2.

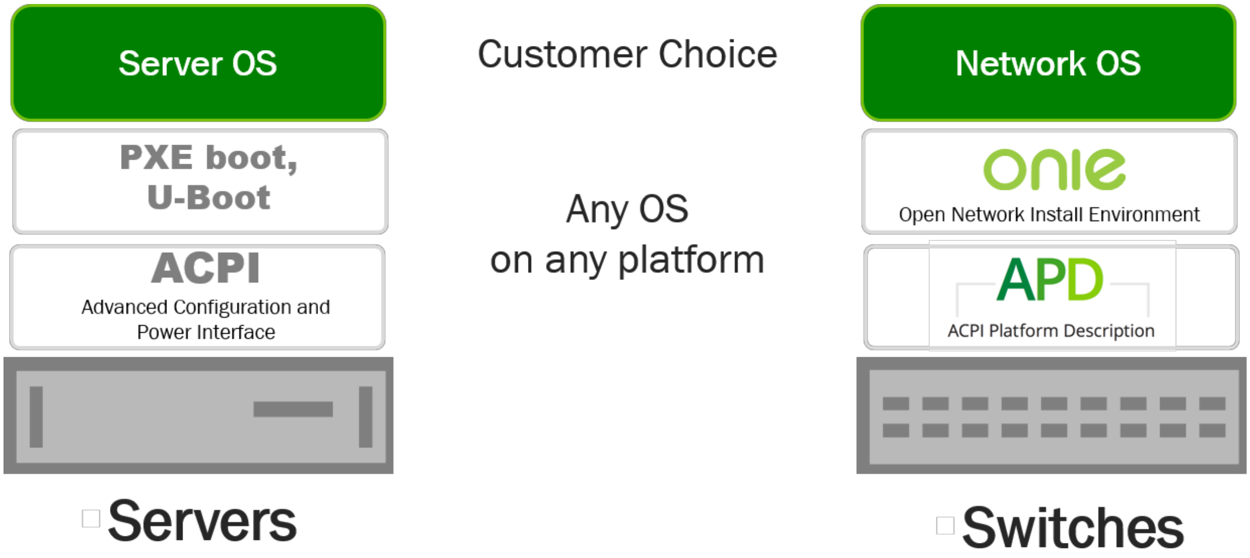

Figure 2 - Analogy between Industries

Design

APD is made up the following:

- Network extensions for the ACPI standard
- Linux kernel code to make use of the ACPI Network extensions
- User-land helpers to the standard isal compiler from ACPICA

One the main motivators for APD is to reduce hardware specific code from the NOS and to place it in the sole control of the hardware vendors using the standard implementation found in servers: BIOS / UEFI. By using this approach illustrated in Figure 3, only generic drivers are needed in the NOS to make use of the underlying hardware which increases adoption time along with portability of both hardware and software. This approach is not just beneficial to Open Networking but traditional, closed networking vendors can leverage this technology to increase platform support.

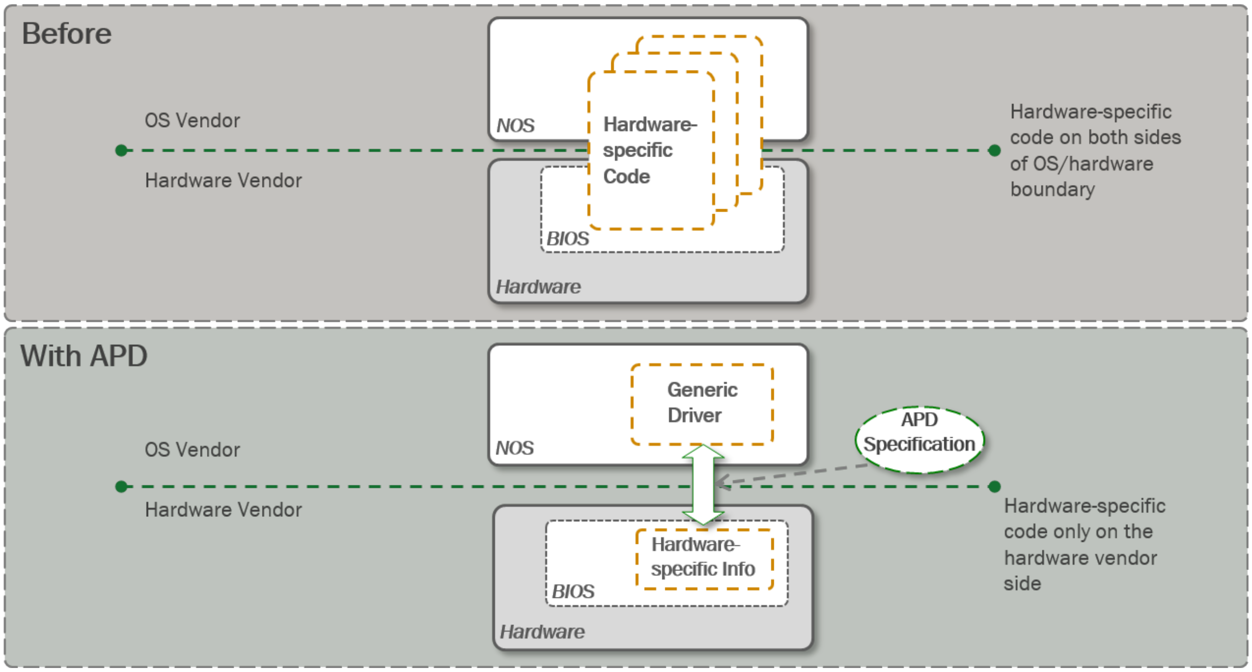

Figure 3 - Design of APD

Test Plan

Testing is a critical component of APD. The project is in the process of developing testing criteria which is occurring over several phases.

1. Compiler checking - syntax and semantic checking by the iasl compiler.
2. Extend fwts (https://wiki.ubuntu.com/Kernel/Reference/fwts) to analyze APD structures.
3. Manual testing checklist (sensors, cable changes, verify config.bcm).
4. Automate the manual checklist.

To provide a common testing platform for developers, we will be packaging an OCP ONIE diagnostic image that is APD aware. This image will provide engineers the basic foundation needed for development and platform testing.

Checklist for Maintenance

We plan on the following maintenance plan to ensure an active and thriving community. Dustin Byford is the project lead (gatekeeper of all commits) to APD. His backups (those that have committer access to APD repositories in the event of his absence) are Curt Brune and Carlos Cardenas. Over the course of the project, individuals that have credible standing with the community and are active with the project will be granted committer access at the discretion of the project lead. If a committer leaves the project or acts in a way that contradicts the goals of the project or becomes abusive to the community, that individual will have their committer status revoked. At no time will the number of committers be less than 3 (1 primary, 2 backups).

In the unlikely event that all committers are incapacitated, it is left to the OCP Foundation to either choose another set of committers (pro-tem or permanent) to lead the project or to disband the project altogether.

Checklist for Governance

This is the list of current governance sites which may change with acceptance into OCP:

- Website: N/A
- IRC: N/A
- Mirror: N/A
- GitHub: https://github.com/CumulusNetworks/apd-tools, https://github.com/CumulusNetworks/linux-apd

- Wiki: https://github.com/CumulusNetworks/apd-tools/wiki

Roadmap

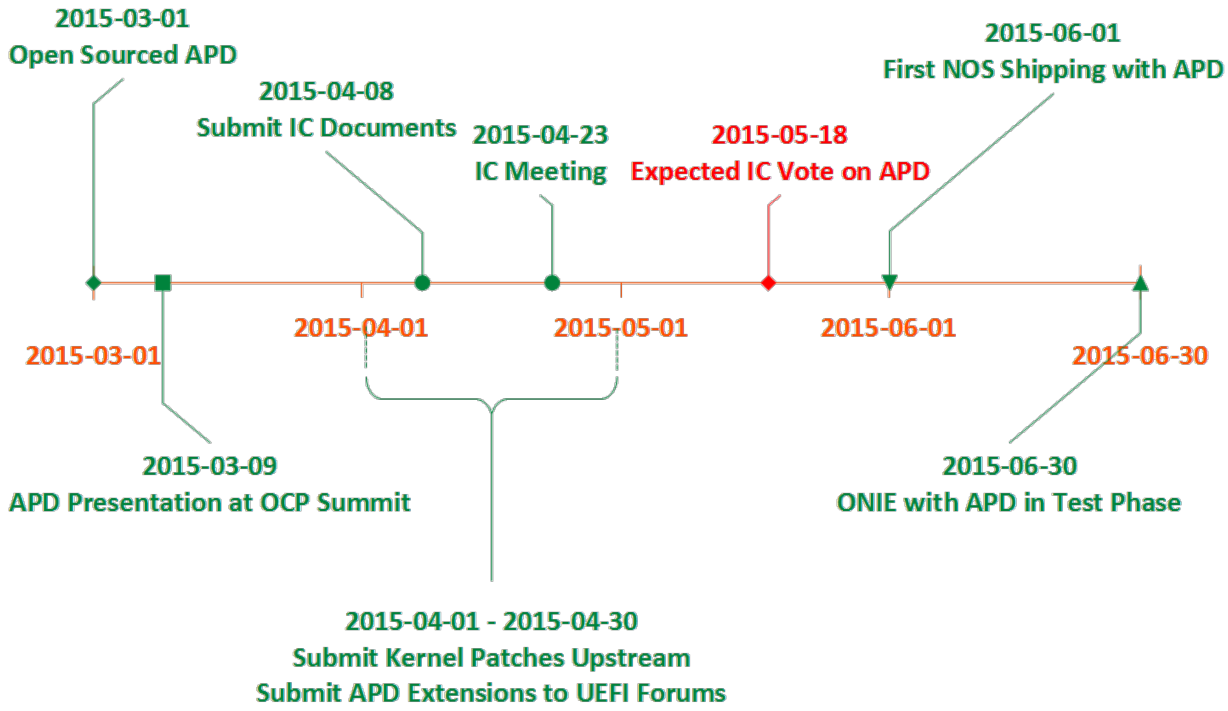

Figure 4 - Roadmap timeline

In the roadmap timeline, we call out key dates for events. Of those, we will highlight the following:

- Linux Kernel submission for 2015 April
- UEFI specification submission for 2015 April
- APD Aware NOS Release
- APD Aware ONIE Release

As APD is open source as of 2015 March 01, it is only a matter of time before NOS vendors utilize APD to support more platforms with ease. APD inside OCP Open Network Linux (ONL) would further increase the reach of ONL's platform support. Furthermore, having OCP ONIE being APD aware allows for additional capabilities inside ONIE such as:

- Implementing thermal controls when booted into ONIE.
- Reporting additional information during discovery process such as switch ASIC and front panel configuration.

○ Can be used in a dynamic fashion in conjunction with SAI to deliver correct SAI library

Supporting Documents

The apd-tools project wiki on github (https://github.com/CumulusNetworks/apd-tools/wiki) is used as a howto guide to help developers new to APD get started. It includes background and technical information with a step-by-step guide to describing a sample platform.